University of Canterbury
Department of Physics and Astronomy

Gravitational Microlensing

An automated high-performance modelling system

by

Alistair M^{c}Dougall

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

November 2014

Declaration of Authorship

I, Alistair M^{c}Dougall, declare that this thesis titled, 'Gravitational Microlensing — An automated high-performance modelling system' and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at the University of Canterbury.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 20 November 2014

"Of course, there is no hope of observing this phenomenon directly."
"Let me also thank you for your cooperation with the little publication, which Mister Mandl squeezed out of me. It is of little value, but it makes the poor guy happy."

Albert Einstein, Publication on microlensing and subsequent letter to the editor
(Einstein, 1936,b)

University of Canterbury

Department of Physics and Astronomy

Abstract

Doctor of Philosophy

by Alistair M^{c}Dougall

Nightly surveys of the skies detect thousands of new gravitational microlensing events every year. With the increasing number of telescopes, and advancements of the technologies used, the detection rate is growing. Of these events, those that display the characteristics of a binary lens are of particular interest. They require special attention with follow-up observations if possible, as such events can lead to new planetary detections. To characterise a new planetary event, high-cadence, accurate observations are optimal. However, without the ability of repeat observations, identification that any event may be planetary needs to happen before it finishes.

I have developed a system that automatically retrieves all microlensing survey data and follow-up observations, models the events as single lenses, and publishes the results live to a web site. With minimal human interaction, the modelling system is able to identify and initialize binary events, and perform a thorough search of the seven dimensional parameter space of a binary lens. These results are also presented live through the web site, enabling observers an up to date view of the latest binary solutions.

The real-time modelling of the system enables a prompt analysis of ongoing events, providing observers with the information, to determine if further observations are desired for the modelled events.

An archive of all modelled binary lens events is maintained and accessible through the website. To date the archive contains 68 unique events' binary lens solutions from the 2014 observing season.

The system developed has been validated through model comparisons of previously published work, and is in use during the current observing season. This year it has played a role in identifying new planetary candidate events, confirming proposed solutions, and providing alternate viable solutions to previously presented solutions.

Acknowledgements

This work would not be possible without the support and help of many people and organisations. I would like to thank the Department of Physics and Astronomy at the University of Canterbury for funding this research. My thanks go to my supervisors, in particular my senior supervisor Dr. Michael Albrow who has helped and guided me throughout my research, and Dr. Orlon Petterson for his continual IT support.

This research is made possible by the essential work of the survey and follow-up teams who tirelessly monitor microlensing events each night. This work has been supported by the generosity of Nvidia Corporation who kindly gifted us state of the art computing facilities, providing new and exciting possibilities.

I am grateful to all those who have taken their time to read through this thesis and help identify any errors, in particular the contributions of Peter Armitage and Ian M^{c}Dougall who gave up their time at short notice to review this work.

Not only are there those who have helped in the development of this thesis, but to all those of Office 805, past and present. Christoph Bergmann, Emily Brunsden, Peter De Vis, Katrin Gressmann, Ariadna Manilla-Robles, Jon Richards, Jeffery Simpson, and Doug Walker, you have all ensured this adventure has been thoroughly enjoyable, making each day fun and exciting.

A delight of working in New Zealand has been the ability to travel the beautiful country, which has been made all the more spectacular by the people I've travelled with, especially Ellen Erikson, Tariq Mcgrane, Rhiannon McNish, and Eddie O'Reilly.

Despite the best planning, the unexpected can always happen. My warmest thanks go to all those who helped us when beginning this journey amidst a natural disaster. In particular the selfless help and hospitality of Jennifer and Matthew Campbell, and the David family, which they continue to provide.

This journey would never have begun without the love and support of our families. My parents Ian and Theresa, brother James, and Emily's parents Mark and Diane, and brother Nathanael. Thank you so much for always being there for us and for traveling to New Zealand for our unexpected wedding, and for the help back in the UK with the originally planned wedding.

Lastly and certainly not least, my biggest thanks goes to my wife Emily who began this adventure with me. Sharing this journey with her has been such a pleasure, and I will be eternally grateful for her ceaseless love and support.

Contents

List of Figures

List of Tables

Abbreviations

AJD	**A**ltered-**J**ulian **D**ate
CCD	**C**harge-**C**oupled **D**evice
CICIRAS	**C**UDA **I**mage **C**entered **I**nverse **RA**y **S**hooting
CPU	**C**entral **P**rocessing **U**nit
CUDA	**C**ompute **U**nified **D**evice **A**rchitecture
DoF	**D**egrees **o**f **F**reedom
DRAM	**D**ynamic **R**andom-**A**ccess **M**emory
EROS	**E**xpérience pour la **R**echerche d'**O**bjets **S**ombres
ESO	**E**uropean **S**outhern **O**bservatory
EWS	**E**arly **W**arning **S**ystem
FLOPS	**F**loating-**P**oint **O**perations **P**er **S**econd
GPU	**G**raphics **P**rocessing **U**nit
IAP	**I**nstitut d'**A**strophysique de **P**aris
ICIRAS	**I**mage **C**entered **I**nverse **RA**y **S**hooting
JD	**J**ulian **D**ate
LCOGT	**L**os **C**umbres **G**lobal **T**elescope **N**etwork
LMC	**L**arge **M**agellanic **C**loud
MACHO	**MA**ssive **C**ompact **H**alo **O**bject
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MicroFUN	**M**icrolensing **F**ollow-**U**p **N**etwork
MOA	**M**icrolensing **O**bservations (in) **A**strophysics
OGLE	**O**ptical **G**ravitational **L**ensing **E**xperiment
PC	**P**ersonal **C**omputer
PLANET	**P**robing **L**ensing **A**nomalies **NET**work
SAAO	**S**outh **A**frican **A**stronomical **O**bservatory

SMC	**S**mall **M**agellanic **C**loud
USD	**U**nited **S**tates **D**ollar
UTC	**C**oordinated **U**niversal **T**ime
WFIRST	**W**ide-**F**ield **I**nfrared **S**urvey **T**elescope

Physical Constants

Speed of Light	c	=	2.998×10^{8}	ms^{-1}
Gravitational Constant	G	=	6.673×10^{-11}	$Nm^{2}kg^{-2}$
Solar Mass	$M_{\odot}$	=	1.989×10^{30}	kg

Symbols

$\odot$	Symbolises the Sun
$\oplus$	Symbolises the Earth
$\hat{\alpha}$	Source light path deflection angle
β	Angular source position
Γ	Limb-darkening coefficient
ΔF	Differential flux
ϵ_i	Fractional mass of the lens objects
ζ	Source position in complex coordinates (in units of θ_E)
θ	Angular image position
θ_E	Einstein radius
$\theta_{m,i}$	Angular lens mass position
$\pi_{\mathbf{E}}$	Two parameter parallax vector
ρ	Finite source size
τ_{Geo}	Geometric time delay
$\tau_{Shapiro}$	Shapiro time delay
τ_{Total}	Total time delay
ϕ	Angle of source trajectory
A	Magnification
A_0	Point source magnification at the finite source centre
D	distance to the source
D_L	Distance to lens
D_{LS}	Distance between source and lens
D_S	Distance to source
d	Binary lens separation (in units of θ_E)
F_0	Reference flux

F_b	Baseline flux
F_i	Flux
F_s	Source flux
I	Image position
I_N	Normalized intensity
$\mathscr{L}$	Likelihood
M	total mass of the lens
M_{pi}	Model's time at peak i
M_i	mass of a single lens object
m_c	Calibration magnitude
m_i	Magnitude
O	Observers position
P_b	Prior probability of any data point being bad
q	Binary lens mass ratio
S	Source position
$\mathbf{s}$	Earth-Sun vector
t_{pi}	Input time at peak i
t_0	Time of closest approach
t_E	Einstein crossing time
t_{com}	Time of closest approach to the mid-point between binary lenses
t_i	A point in time
t_{mp}	Time of closest approach to the mid-point between binary lenses
t_p	Fixed point in time
u	Distance between the source and lens in the lens plane (in units of θ_E)
u_0	Distance of closest approach (in units of θ_E)
u_{com}	Distance of closest approach to the centre of mass (in units of θ_E)
u_{mp}	Distance of closest approach to the mid-point between binary lenses (in units of θ_E)
V_b	Mean of a bad data distribution
$\mathbf{v_p}$	Velocity vector of the Earth, at time t_p
Y_b	Variance of a bad data distribution
z	Image position in complex coordinates (in units of θ_E)
z_{m_i}	Lens mass position in complex coordinates (in units of θ_E)

Dedicated to Biscuit, for her continual care and support that never wavered, even through the shaky (earthquakes) start, so far from home.

Chapter 1

Introduction

1.1 Overview

Gravitational microlensing is a relatively young area of research, yet it was initially considered as far back as 1704 by Sir Isaac (Newton, 1704) who asked *"Do not bodies act upon light at a distance and by their action bend its rays, and is not this action strongest at the least distance?"*. It was later theorized by Johann Soldner in 1804 (von Soldner, 1804) and more recently described in 1915 by Albert Einstein's General Theory of Relativity (Einstein, 1915). It is a phenomenon that enables measurements which can detect exoplanets in a unique mass to orbit ratio, a ratio that may lead to the discovery of habitable Earth-like planets.

1.1.1 Microlensing — A method to detect another Earth

A gravitational microlensing event occurs on the rare occasion when a star observed from Earth, the source star, moves behind another massive object in space, the lens. The gravity of the lens causes the light from the source to bend, magnifying the brightness observed from Earth. Such magnifications can be up to 1000 times the original brightness, with a standard event lasting from a few days up to a month (Paczyński, 1991; Kiraga & Paczyński, 1994; Sumi et al., 2011). By studying such events it is possible to help answer questions about the universe around us, such as "why is the galaxy as it is?", "how did the Earth form?", and "is it unique?". Many theories have been proposed to unravel these mysteries, but still no conclusive answers exist. Microlensing measurements can help move us a step closer to providing some answers. These measurements play a key role by filling the gap in the library of data used to statistically model the galaxy, detect and study dark matter, and detect exoplanets.

Studies of microlensing events can be made by observing the variation in brightness of the event over time, producing a light curve. A single lens object will produce a very distinct light curve shape known as a Paczyński curve (single lens model). Deviations from this model can be attributed to binary star systems or even planetary systems. An example of this is shown in Figure 1.1. Intense observations of microlensing events allow small deviations from the single lens model to be detected. With a thorough data set, accurate models can be made to fit the data, models that can indicate if a new planet has been found.

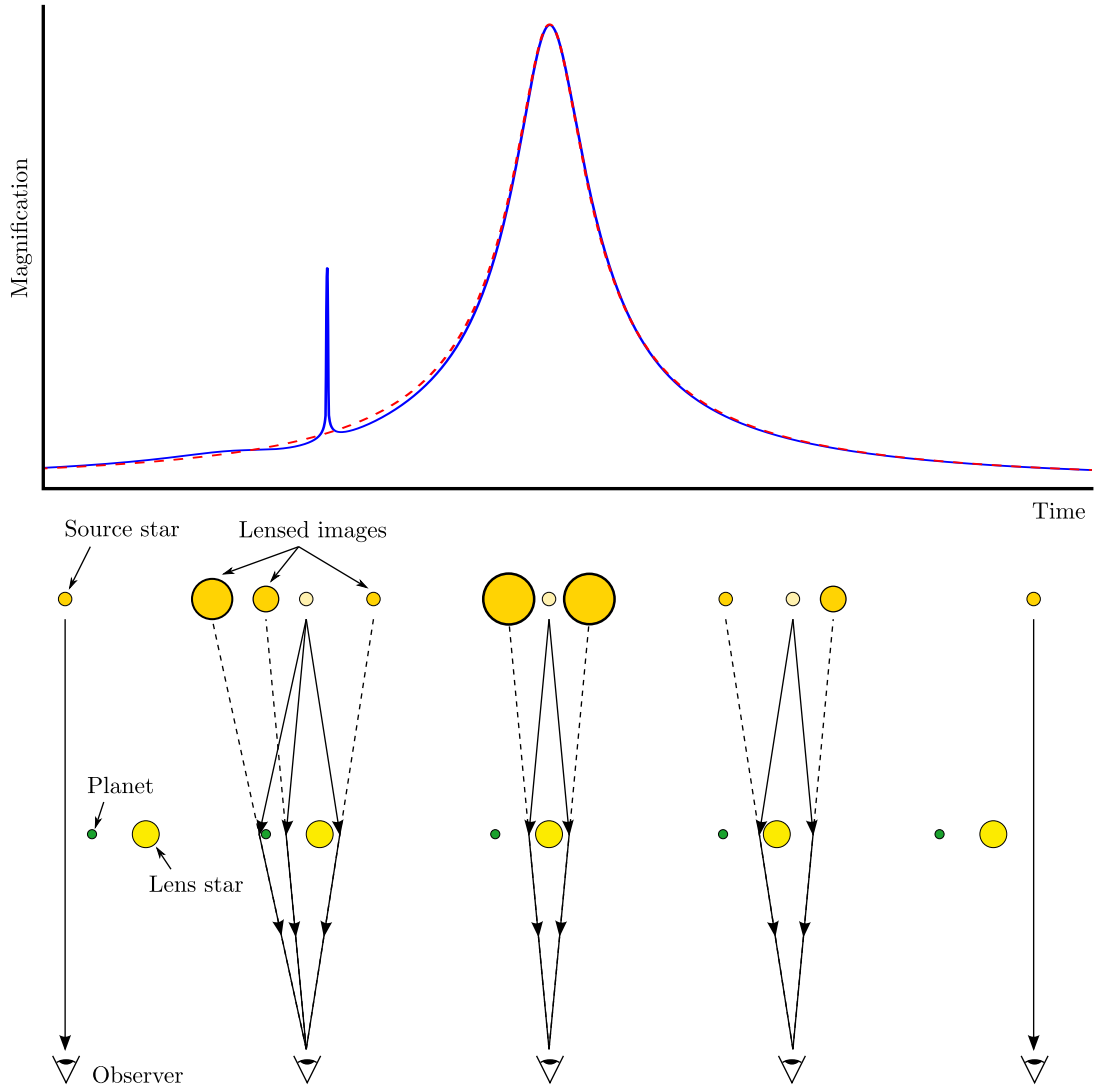


FIGURE 1.1: Graphical representation of how the source is lensed, causing an anomaly from a planetary binary lens. Comparison of the planetary light curve (solid blue) and the single lens light curve (dashed red) shows the clear deviation from the single lens model. The bottom half of the figure represents the contributions of light from the lensed images (top row), as a binary lens (middle row) moves through the line of sight of the observer (bottom row), resulting in the magnifications seen in the light curve above.

Using microlensing to detect planets is not without its problems. Not only does it rely upon the rare occurrence of a star alignment, meaning repeat measurements are very improbable, but it is also only possible to determine a rough distance estimate to the exoplanet from Earth. These reasons however, should not deter us from continuing to use this technique; experimental methods can enhance our chances of observing these events, such as observing for long periods of time in a dense star field (the galactic bulge), which greatly increases the frequency of event detection. Not all microlensing events are caused by binary lenses, and single lenses also provide crucial information for statistical modelling of the planet frequency throughout our Galaxy.

The microlensing method is not the only way to search for exoplanets. To date, from all exoplanet search methods a total of 1810 exoplanets have been identified, according to the The Extrasolar Planets Encyclopaedia[1] on 2014 July 10. Other approaches, such as transit detections and radial velocity measurements, are both very successful exoplanet hunting techniques but they are not able to provide all the answers on their own.

The formation of planets is thought to be a common outcome during the creation of a new star. A star will form due to the gravitational collapse of a large molecular cloud, and because of angular momentum this cloud collapses onto a disk in a plane that is orthogonal to the cloud's total angular momentum. The new star will slowly accrete the majority of this molecular cloud, but it will not use it all. The commonly accepted solar nebula theory details that from the remaining dust cloud, planetary bodies are able to form (Perryman, 2011). By the same concept of the star forming due to gravitationally accreting the surrounding dust, rocky bodies are able to form. This process will continue with larger bodies as they collide and stick together, ultimately leading to protoplanets. The formation of terrestrial planets via this method is commonly accepted, but it is insufficient to explain the existence of the giant planets. To date, the two main theories to explain giant planets are, core accretion and gravitational instability.

Core accretion is a commonly accepted theory which is able to explain a wide range of planetary results, with the high and low elemental mass abundances (Mizuno et al., 1978; Mizuno, 1980), and the ice and gas giant planet formation on similar time scales to the nebula gas lifetime (Perryman, 2011). The theory explains that by the same process as above, large cores ($5-20M_{\oplus}$) can form in the protoplanetary disk. However, simulations have shown that there are typically not enough solids in the inner protoplanetary disk to achieve this. This situation is resolved if the large planetismals form at larger radii, beyond the snow line. The increased distance reduces the gravitational influence from the host star, and it is cold enough for water and other molecules to condense, increasing the amount of matter that can be accreted. Once these cores reach critical mass

[1] http://exoplanet.eu/

(Bodenheimer & Pollack, 1986; Pollack et al., 1996), the rate of planetismal accretion is exceeded by gas accretion. This gas accretion rate grows exponentially, and further accreted planetismals are unable to survive, breaking up from the gas drag and pressure, which contributes to the heavy elements in the gas giants atmospheres. The formation of giant planets are very sensitive to the initial conditions. Smaller protoplanetary cores take longer to form, which can result in the protoplanetary gas being dispersed before they are able to complete their gas accretion, which leaves behind an ice giant. Although this theory provides an apparently good explanation for giant planet formation investigations are still on going, especially to do with formation time scales. With research looking into young solar like stars dissipating their disk of gas before heavier gas giants can form, or if ice giant formation requiring too much time to form.

Gravitational disk instability in contrast, is a much simpler one step process that can explain the formation of some giants (Kuiper, 1951; Boss, 1997; Durisen et al., 2007). The advantages of gravitational instability are that it does not rely on any solid body accretion, and that is forms in short time scales. The theory uses the undisputed fact that at some point in time, the protoplanetary disk will be gravitationally unstable. These instabilities could lead to spiral arms in the disk that result in forming self-gravitating regions that lead to giant protoplanets.

With two very different theories, a lot of research has focused on determining which is more likely to be true. Evidence is presented that supports the core accretion (Mordasini et al., 2008), and gravitational instability (Boss, 2001) theories, with others stating that the current evidence supports a mix of both theories (Matsuo et al., 2007).

The transit detection method is a common technique used to detect exoplanets. It involves observing a star and looking for a decrease in the star's flux (ΔF) caused by a planetary body passing in front of the host star, blocking out some of the light. By repeatedly measuring the change in flux during a transit, it is possible to determine the radius of the planet and the period of its orbit (Seager & Mallén-Ornelas, 2003). A selection bias exists, where a smaller period and therefore closer separation planet is more likely to be detected due to a complete orbit occurring in a shorter time, and the larger a planetary body and the smaller the host star, a greater fraction of light is obscured, causing a larger flux change. With the recent success of the Kepler programme, transit detections make up the largest contribution of exoplanet detections with a total of 1140 confirmed planets.

Transit detections methods have created a lot of news recently (Moutou et al., 2013; Kipping et al., 2014), with the Kepler program proposing thousands of new planetary candidates (Rowe et al., 2014; Lissauer et al., 2014), and questioning our current understanding of planetary formation (Lissauer et al., 2014). Yet instead of answering all our

questions about exoplanets, it has presented several more. Such as planetary spin-orbits being misaligned (Addison et al., 2014), and the detections of large numbers of giant planets orbiting close to their host star has made us further consider, how do they get there (Lissauer et al., 2014)? The transit method not only allows us to determine the radius and orbital period of an exoplanet, but it also enables us to study the planets atmosphere (Venot & Agúndez, 2014). This provides additional information, which can be used to further test out planetary formation theories (Madhusudhan et al., 2014). The transit method has been able to detect many short period targets, but requires a long observing program to perform multiple measurements of long orbits (large orbital radius), a region of planetary space that is left to gravitational microlensing to explore.

Radial velocity measurements have a similar sensitivity region as the transit method, and this is the second most successful method of planet detection. It relies on measurements of the orbital component of the host star about the star-planet centre of mass. A repeating change in movement of a star can be seen in the Doppler shift of its spectral lines, from which it is possible to infer the minimum planet mass and several orbital parameters of the system. This technique is therefore more sensitive to larger mass planets and shorter orbital periods as these cause the greatest movement of the host star's orbit. There is also a selection bias for shorter period orbits as they do not need to be observed for as long before a complete period can be seen. Orbits around low mass host stars are also easier to detect, as the lower the stellar mass the larger the movement caused by the gravitational pull of a planetary body. The radial velocity measurements contribute 574 of the planetary detections, and is currently the main method used to look for an Earth mass planet in our neighbouring binary star, α Centauri (Dumusque et al., 2012; Endl et al., 2014b; Bergmann et al., 2014).

The radial velocity method uses high-resolution spectroscopy to detect the small Doppler shift of the host star (Mayor et al., 2003). This measuring technique makes it technically challenging to perform a wide field survey, as well as being limited to bright host star targets (Campbell et al., 1988; Patel et al., 2007; Mayor et al., 2003). Although it is harder to detect new planets without a survey, a long observing schedule and focused observations has shown that this method is successful in providing interesting new results. The radial velocity detection method has discovered multiple planetary systems (Wittenmyer et al., 2014), even around a binary pair (Desidera et al., 2014), and the closest brown dwarf-stellar binary found to date (Jones et al., 2014). This method has also been applied to planet searching specifically around post main-sequence evolved stars (targets that are frequently avoided), to ensure a full overview of planetary statistics (Niedzielski et al., 2014).

Radial velocity exoplanet detections are continuing to provide useful information about planetary statistics on its own, however combined with the transit detection method, a greater amount of information can be inferred (Endl et al., 2014a). Using the large number of candidate detections from a transit survey, such as Kepler, follow up radial velocity measurements are able to provide additional constraints on the planetary parameters. This approach is also able to determine additional planets in a system, which do not undergo a transit and therefore have previously been missed.

Both transit and radial velocity measurements contribute a lot to planetary statistics, however they are both limited to detecting large planetary bodies with short orbital periods. Not only this, but as they measure the light from the host star, they are restricted to bright sources, which are typically located in the local neighborhood.

For objects that have a periodic signal such as eclipsing binaries, pulsating stars and pulsars, precise measurements of the signal's timing can identify deviations in the period, which can be caused by movements of the signal source. As with radial velocity measurements multiple objects orbiting a centre of mass results in the signal source moving, but this time instead of looking for Doppler shifts, the change in the signal's period is measured. As the source is moving towards and away from the observer, it means the signal has to travel a shorter or greater distance, therefore changing its period. The precise measurements of a 6.2ms pulsar lead to the discovery of the first two exoplanets by Wolszczan & Frail (1992). Further investigations of this planetary system lead to the detection of a third exoplanet, which is the smallest mass exoplanet currently known at $0.02M_{\oplus}$, which is comparable to the mass of the Moon (Wolszczan, 1994a,b; Konacki et al., 1999; Wolszczan, 2008). Despite these impressive achievements, to date this method has contributed the fewest number of exoplanet detections, with a total of 15.

An alternate method of detecting exoplanets is through direct imaging. This requires taking very high-resolution measurements and identifying the position of an exoplanet by the light it reflects off its surface, or infrared emissions from the planet itself. This method relies on precise observations and therefore requires large telescopes and advanced techniques, such as nulling interferometry (destructive interference of the host stars emissions), to avoid as much noise as possible. This planet hunting method faces numerous technical challenges in the development of observational methods, which results in a sensitivity to larger mass planets at large distances from their host stars, and stars that are very close to Earth. Yet the technical difficulties in performing such precise measurements have not prevented the detection of exoplanets. To date, 50 exoplanets have been found through direct imaging.

Astrometry uses the same multiple body properties, that a system will orbit about its centre of mass, as radial velocity and timing methods use. Instead of measuring an effect that any movement will cause, such as Doppler shifts or signal period delays, astrometry tries to measure the actual position of the star and determine its orbital movement on the plane of the sky. Detecting exoplanets via orbital motion results in the same selection bias and sensitivity as radial velocity and timing. However, as it determines the orbital parameters on the plane of the sky, it complements the other methods that determine the orbital motion into the plane of the sky. Requiring the precise measurements of a star's position means performing a large sky survey is technically challenging, which results in there being no exoplanet detections via this method to date. However, with the launch of the Gaia observatory at the end of last year, several thousand new exoplanet candidates are expected to be discovered through this method.

As shown above, each method has strengths and weaknesses in what information they can infer about a planetary system, and what planetary bodies they are more sensitive to. Unlike the methods that are most sensitive to large mass planets close to the parent star, microlensing is sensitive to lower mass objects with larger orbital radii, as Figure 1.2 shows. Microlensing is also unique in that it is the only method able to detect planets that are very large distances from the Solar system because the method to detect new planets is not dependant on luminosity, but only upon the lens' mass. This important dependence is why microlensing is more sensitive to very distant objects and small planetary masses, which has been shown by Paczyński (1996), who stated that it is even possible to detect moon mass objects using ground based telescopes. With a suitable experimental approach, this technique can be extended even further, as shown by Ingrosso et al. (2009, 2011) who states that it is possible to detect exoplanets down to $20M_{\oplus}$ in our neighbouring galaxy, Andromeda, M31.

These important differences have resulted in a wide range of new discoveries coming from microlensing observations. One of the most notable of these was the announcement of the detection of free-floating planets (Debes & Sigurdsson, 2007; Sumi et al., 2011). These cold objects are, as the name suggests, planetary mass bodies that have been found with no host star close by which they may be orbiting. Another planetary body, which microlensing has detected is the existence of cold planets at wide separations. Not only was microlensing the first method to detect both of these objects, but it has also provided evidence to suggest that both are common within our galaxy (Gould et al., 2006; Beaulieu et al., 2007; Kubas et al., 2008; Sumi et al., 2011).

The fundamental difference between microlensing and the other previously mentioned methods of exoplanet detection, is that microlensing does not need to observe any emissions from the lens. This enables the method to be used to detect cold, dark objects.

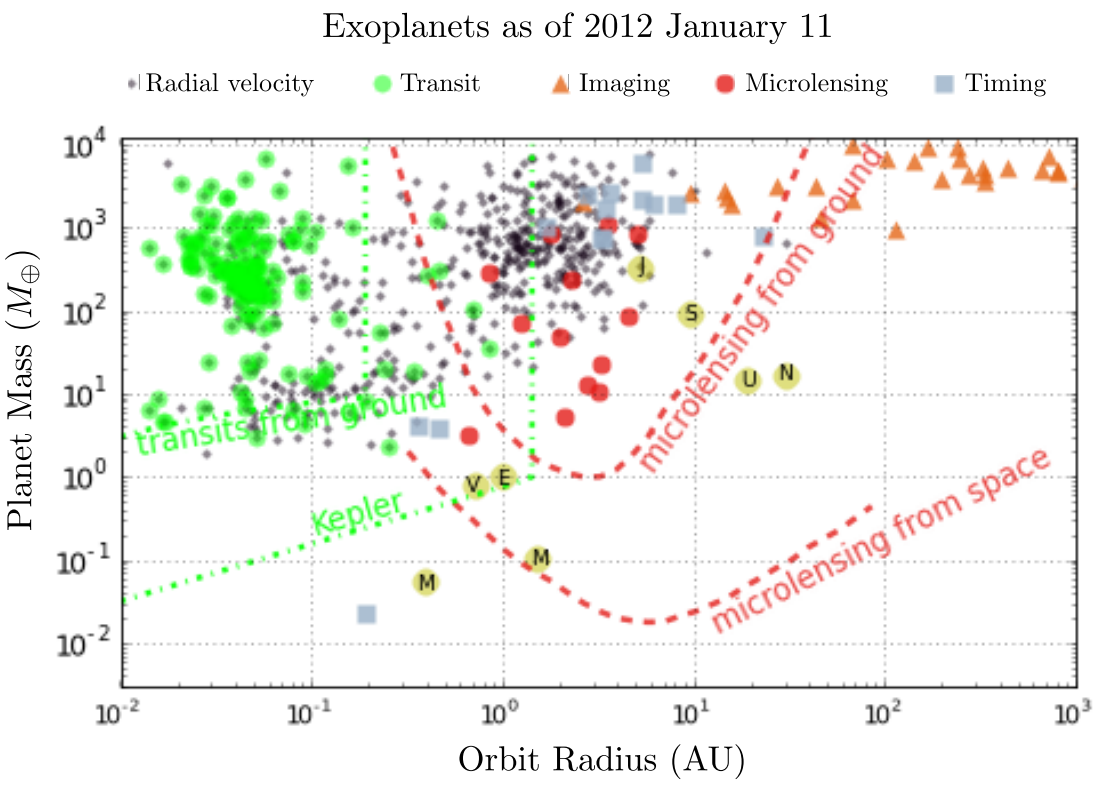


FIGURE 1.2: A depiction of the mass, separation ratio which exoplanet hunting techniques are sensitive to. The bold capitalised letters correspond to the planets in our solar system, for comparison. This figure was made using the data from The Extrasolar Planets Encyclopaedia, and has been modified from the original presented by the RobotNet group (http://robonet.lcogt.net/research/ on 2014 July 11)

As microlensing takes a snapshot of the planetary system and does not rely on observing orbital periods, it is much easier to detect wide separation planets, which have long orbital periods. The unique mass/separation ratio, that microlensing is sensitive to, provides important information which helps develop our understanding of overall planetary abundances (Cassan et al., 2012), and refine planetary formation theories. An example being the work of Kubas et al. (2012), who showed that even around very low mass stars, large frozen Earths can form. These cold, dark, and large orbital planetary bodies are only being detected through the microlensing method, as they are outside the detection range of other methods (Jung et al., 2014). Through these planetary detections, which are only accessible with microlensing, the first Uranus-like planet has been discovered by Poleski et al. (2014). In addition to the discovery, they comment on how such a detection prompts for the careful consideration of free-floating planet abundances.

Not only does microlensing provide new discoveries of cold distant planets, but it has also been used to detect a terrestrial planet around one member of a stellar-binary pair (Gould et al., 2014). Other microlensing detections have shown large planets that

possibly lie within the habitable zone, which could in principle host habitable moons (Batista et al., 2014), planetary systems of giants orbiting dwarf stars, and possible free-floating giant planets with a moon (Batista et al., 2011; Bennett et al., 2014).

With the increasing number of telescopes, our understanding of planetary abundances will grow. The Korean Microlensing Telescope Network, which is set to be operational in 2015, is predicted to detect more than 30 planetary events per year (Henderson et al., 2014). Where as the Euclid space based observatory has been predicted to detect hundreds to thousands of free-floating planets per month (Hamolli et al., 2013). Not only does microlensing support planetary detections but, as stated by Evans (2014), combined with Gaia observations, it is the best way to investigate the masses of brown dwarfs in our solar neighbourhood.

With no one method able to detect the whole range of possible planetary bodies, the unique sensitivity of the exoplanet detection region using microlensing, provides a complementary technique when studying the overall planetary population statistics. Progress has been made by all of these methods, helping provide vital results to determine planet population statistics, with an overlap in the detection sensitivity of these methods enabling a validation of the population results determined Dominik et al. (2010). Due to the sensitivity region of the microlensing technique, it is likely that it will be the first method to detect an Earth-mass exoplanet. This likelihood will be dramatically increased in the future with the proposed WFIRST (Wide-Field Infrared Survey Telescope) space mission, which will provide dedicated microlensing observations.

1.2 The current situation

With the growing network of telescopes and technological improvements made, the number of microlensing events detected in the hundreds of millions of stars observed nightly is growing each year. In 2012, a total of 1998 unique events were monitored, 2013 identified 2220, and currently the 2014 season has found 1556[2]. This vast amount of data needs to be regularly monitored and modelled so that interesting events can be identified as early as possible, enabling follow up observations to be made. Real time data reduction is performed at each of the survey telescopes, and early warning alert systems (Udalski et al., 1994; Alcock et al., 1996; Glicenstein, 2001; Bond et al., 2001; Udalski, 2003) have been developed to identify a microlensing event and publish the data on the survey group's web sites. A similar system has been developed by Dominik (2007), where real time modelling of single lens events is performed so that anomalies from the single lens model can be found in real-time. He states that such a system is vital to detect

[2] As of 2014 July 11

low mass planets as they can cause short subtle deviations from the single lens model, potentially lasting only a few hours (Mao & Paczyński, 1991), and require instant high cadence observations to capture (typically every 10-15 minutes) (Dominik et al., 2008; Dominik, 2008; Dominik et al., 2010). Alternate microlensing search strategies exist that do not focus on searching for potential anomalies in the shoulders of a single lens event, but instead perform intense follow up observations of high magnification events, which are more sensitive to planetary deviations (Gould et al., 2010). Both of these methods rely on real-time modelling of the single lens events, so that they know what targets to focus their follow up observations on. With a limited number of follow up telescopes available and an ever growing number of event detections, accurate real-time modelling of each event is becoming more important to help direct the telescope's observing time. This problem is extended further to situations where multiple events are known to be undergoing anomalous behaviour, and a preference needs to be made about which event to follow up. A real-time binary lens modelling system is therefore required to identify anomalous events and model them, providing observers with detailed information about the cause of the anomaly and any possible future deviations that should be expected. Such a system could help distinguish between two different anomalous events where one may be caused by a binary star system, and the other due to a planet.

The complexity of binary lens systems hinders the ability to achieve real-time modelling, and with a growing number of anomalous event detections a time critical analysis of all anomalous events is required. The development of a rapid modelling procedure able to explore a wide parameter space and determine viable binary lens solutions, is essential to helping prioritize anomalous targets for follow up observations.

1.3 A proposed solution

Since microlensing surveys first began, significant advances have already been made in real-time modelling. Single lens models originally took anywhere from a few months to three years before they were announced (Udalski et al., 1994), however, with the current use of Early Warning Systems (EWS) (Bond et al., 2001; Udalski, 2003), survey groups now publish their microlensing detections in real-time. This can be further enhanced by a real-time multiple lens modeller, able to estimate a range of possible causes of anomalous events suggesting the most likely solution. The benefits of real-time anomaly detections, and modelling have already been shown by Bozza et al. (2012) and Ryu et al. (2010). With a fast, automated system, it would be possible to see the current situation of all ongoing anomalous microlensing events, which can be used to help direct follow up telescopes.

An overview of the mathematical derivation of microlensing is presented in Chapter 2. This chapter outlines multiple methods used to model microlensing events, and describes several additional effects that are to be considered upon modelling microlensing events. A list of the main observatories that contribute microlenisng data is presented in Chapter 3, detailing their operational history and what equipment they use. Data from each observatory is provided in various forms, meaning they cannot all be treated the same. Therefore a detailed methodology of how each site's data is used can be found in this chapter. The use of Graphical Processing Devices (GPUs) is investigated as a method to model multiple lens events. Details of their architecture, strengths and weaknesses, and suitability to certain problems are shown in Chapter 4. The unique nature of these devices enables alternate ways of performing common computational tasks, some of which have been detailed in this chapter. The development of new modelling methods are detailed in Chapter 5, where an outline and mathematical methods used to model a single lens event are shown. This is expanded further with the presentation of a new search strategy for binary lens models using the advantages of GPU devices. As a way of visualising and controlling the modelling of single and binary lens events, a web site and control software have been developed which are presented in Chapter 6. With use of the methods presented in these chapters, the results of several past events are presented in Chapter 7, these are shown to validate the new method before it is used to analyse new un-published microlensing events from the current observing season in Chapter 8. Finally, all the work presented in this thesis is summarised in Chapter 9, before a list of appendices are presented that detail the mathematical derivations, and visual representations of microlensing coordinate systems and computational programs.

Chapter 2

Microlenses

2.1 Origins

In 1986, significant progress was made in the research field of gravitational microlensing through the work of Bohdan Paczyński (1986b,a), who build upon previous studies, such as those performed by Liebes (1964) and Refsdal (1964). His work showed that not only was it possible to model a microlensing event of stars within our own galaxy and the Magellanic cloud, but also with a suitable experimental approach, it would be possible to observe such an event. Despite the limited technological availability, he proposed a potential observing programme to last two years that should detect a microlensing event due to a $100M_{\odot}$ star. The goal of the programme was to study the theorised dark matter component of our galaxy, by searching for MAssive Compact Halo Objects (MACHOs), and also possibly detect events caused by objects in the range $10^{-6} < M/M_{\odot} < 10^{2}$. Paczyński recognized such an observing programme would produce a vast set of data requiring analyses which would be difficult, but he commented on work being undertaken to automate some of the data process.

Not long after Paczyński published his paper the MACHO collaboration (Alcock et al., 1993), Expérience pour la Recherche d'Objets Sombres (EROS) (Aubourg et al., 1993), and Optical Gravitational Lensing Experiment (OGLE) (Udalski et al., 1992; Szymański et al., 1993), began with the goal of detecting the dark halo objects by searching for these rare microlensing events. Since then more projects have been established to search the skies for these events (Abe et al., 1996; Albrow et al., 1996; Alard, 1996), and the observational focus has changed in favour of detecting a range of exoplanets towards the galactic bulge. Improvements in telescope technology and the increase in observations has lead to an even larger quantity of data available for analysis. Automated data

reduction processes are now essential to process this vast quantity of data, with continual work performed to improve automated microlensing modelling techniques.

2.2 Derivation

All microlensing modelling stems from the most basic example of a point source point lens model. From the schematic shown in Figure 2.1, as a source star passes behind a lens, the gravity from the lens bends the light from the source star. This produces an image at point I, which has been deflected by an angle $\hat{\alpha}$. The image position can then be solved by considering Fermat's Principle (Hecht, 2002), "A light ray in going from point S to point O must traverse an optical path length that is stationary with respect to variations of that path". This effectively states that the image will be where the light beam takes the least time to traverse; these image positions correspond to the stationary points of the time delay surface (Equation 2.3).

2.2.1 Time delays

To determine the time delay surface, two contributing factors need to be considered, the geometric time delay (Equation 2.1) and the gravitational time delay (Shapiro delay) (Equation 2.2) (Gaudi, 2011).

The geometric time delay takes into account the increased path length that the light from the source has to travel, as opposed to if it were traveling in a straight line. Included in this effect is the increased path distance due to the curvature of space, caused by the mass of the lens,

$$\tau_{Geo} = \frac{\Delta t}{\Delta t_0} = \frac{1}{2}(\theta - \beta)^2 \quad . \tag{2.1}$$

The second effect is the gravitational time delay, this is caused by the source light passing through the gravity field of the lens, which has the affect of slowing it down, producing a delay of,

$$\tau_{Shapiro,i}(\theta) = -\frac{4GM_i}{Dc^2} \ln|\theta - \theta_{m,i}| \quad , \tag{2.2}$$

where $D = \frac{D_S - D_L}{D_S D_L}$. The resultant total time delay is simply the combination of the two equations 2.1 and 2.2,

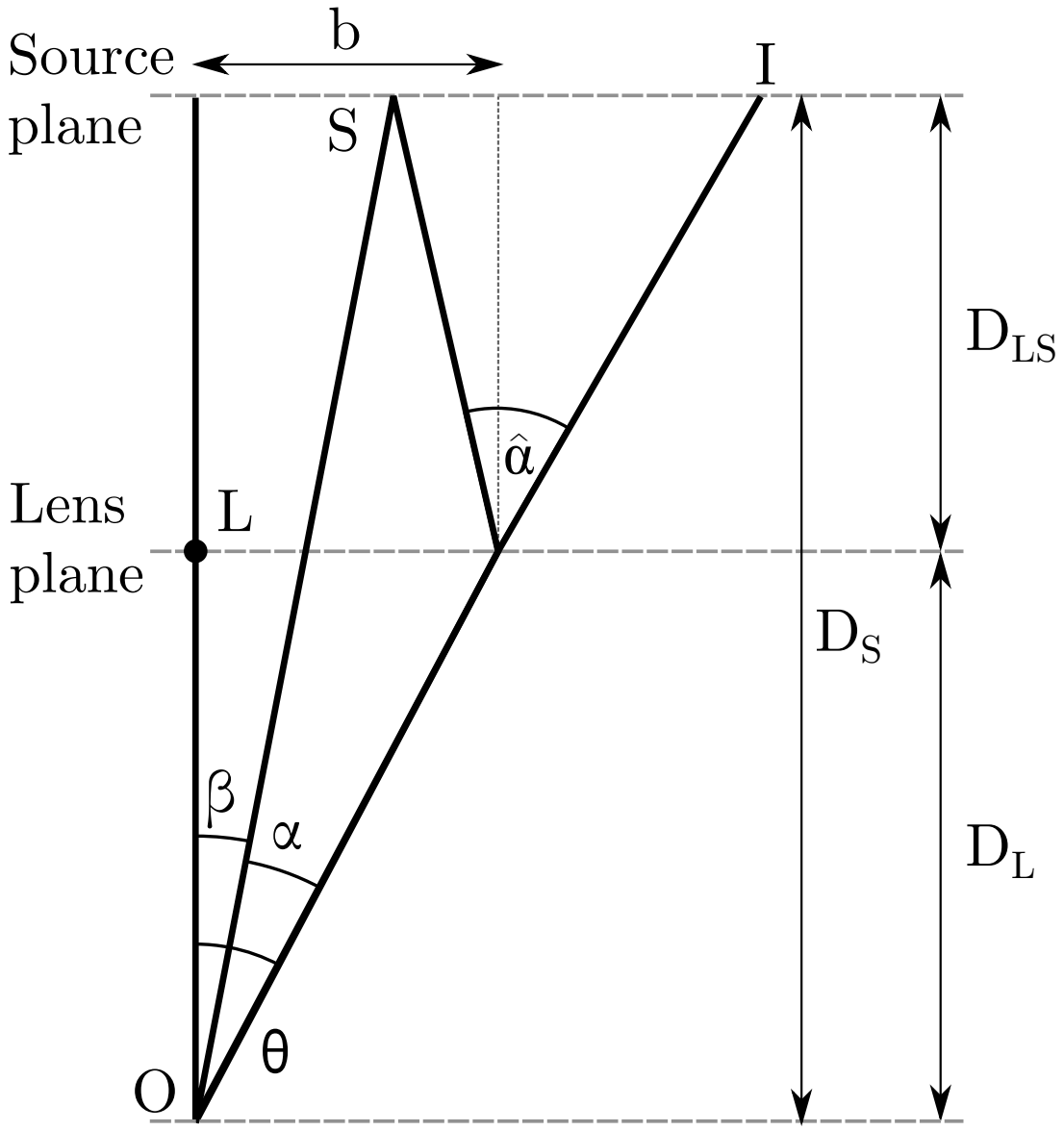


FIGURE 2.1: The geometry of the path the light travelled from the source (S) being microlensed by a dense object L, a distance D_L away, in the lens plane.

$$\tau_{Total}(\theta) = \frac{1}{2}(\theta - \beta)^2 - \sum_{i=1}^{N} \frac{4GM_i}{Dc^2} \ln|\theta - \theta_{m,i}| \ , \tag{2.3}$$

where N is the number of lenses.

2.2.2 Image positions

As stated by Fermat's Principle, the images will only be created at the maxima, minima and saddle points of the time delay surface. It is also noted that additional images to

the original single image are always created in pairs, meaning that there will always be an odd number of images $(2N + 1)$.

The image positions can therefore be calculated by taking the differential of the total time equation (Equation 2.3) with respect to the image position and equating it to zero, the stationary points. Re-arranging for the source position β leads to the general form of the lens equation for point masses,

$$\beta = \theta - \sum_{i=1}^{N} \frac{4GM_i}{Dc^2} \frac{\theta - \theta_{m_i}}{|\theta - \theta_{m_i}|^2} \quad . \tag{2.4}$$

2.3 The single lens

Consider the case of a single lens, N=1, which produces the following equation, showing that circular symmetry is valid for images along the axis of the source and lens ($\beta = 0$),

$$\beta = \theta - \frac{4GM_i}{Dc^2} \frac{1}{\theta} \quad , \tag{2.5}$$

defining the Einstein radius to be,

$$\theta_E = \sqrt{\frac{4GM}{Dc^2}} \quad . \tag{2.6}$$

If all angles are scaled by the Einstein radius, and

$$u = \frac{\beta}{\theta_E} \quad , \quad y = \frac{\theta}{\theta_E} \quad , \tag{2.7}$$

where u and y represent the source and image position in the lens plane respectively, we obtain from Equation 2.5,

$$u = y - \frac{1}{y} \quad . \tag{2.8}$$

This is the simplified lensing equation for a single point mass lens, from which a magnification can be calculated.

2.3.1 Single lens magnification

Liouville's theorem, first shown in 1902 (Gibbs, 1902), stated that lensed packets of light conserve the brightness they would have if un-lensed. This principle means that the determinant of its Jacobian matrix can be used to express the factor that the function expands or shrinks its volume (Schneider et al., 1992). As the magnification is a ratio of brightness of the lensed images and the un-lensed image, taking the inverse of the Jacobian matrix determinant of equation 2.8 leads to the magnification of the lensed source star (for a full derivation see Appendix A),

$$A = \frac{u^2 + 2}{u\sqrt{u^2 + 4}} \quad . \tag{2.9}$$

At any given time, t_i, a value for u, the distance between the source and lens projected in the lens plane can be calculated using trigonometry and the source star's relative velocity,

$$u = \sqrt{u_0^2 + \left(\frac{t_i - t_0}{t_E}\right)^2} \quad , \tag{2.10}$$

where u_0 and t_0 are the separation and time at closest approach respectively, and t_E is the Einstein crossing time (for a full derivation see Appendix B). With equations 2.9 and 2.10, a single-lens model can be produced for fitting to various sources of data.

2.4 The binary lens

Now consider the case of a binary lens. Starting from the general lens equation (2.4) and setting the number of lenses to be two (N=2), it is possible to determine the binary lens equation. However, as it is no longer circularly symmetric, additional steps are made to reduce it down to a simplified form, as shown in Appendix C. This leads to a general form of the binary lens equation in complex coordinates

$$\zeta = z - \frac{\varepsilon_1}{\bar{z} - \bar{z}_{m_1}} - \frac{\varepsilon_2}{\bar{z} - \bar{z}_{m_2}} \quad , \tag{2.11}$$

where ζ and z are the source and image positions in complex coordinates respectively, z_{m_i} the lens mass position, and ϵ_i the fractional mass of the lens objects.

2.4.1 Binary lens magnification

As for a single lens, to determine the magnification a ratio of lensed and un-lensed image brightness can be used. This first requires the solutions to the binary lens equation so that the images can be determined. Unlike the single lens case, the image positions are solutions to binary lens equation (2.11), which are not trivial to find. This equation can be re-arranged so that the coefficients of a 5^{th} order polynomial are found, allowing numerical solving methods to be used to determine the solutions of the equation, as detailed in Appendix D. Using a set of true solutions (either three or five), a Jacobian matrix can again be used to determine the magnification from each image (As shown in Appendix E). These are then summed together to provide the total magnification of a binary-lensed point source. This can be used in the same way as with a single lens, to test how well a model fits the data.

2.5 Microlensing modelling techniques

Microlensing causes multiple images to be formed around the lensing object, which results in the magnification of the source. As the name suggests, the size of the lensed images and their separations are very small, too small to be seen individually through a telescope. The only information that can be observed is the total brightness of a lensed event. Therefore modelling is required to determine how the source and lens interacted and produced the change in brightness that is observed.

A single lens event is easily modelled by analytically solving the single lens equation (2.9), however a binary lens cannot be treated in such a simple way. Therefore several methods have been developed to model binary lens events, taking into account accuracy, speed, and higher-order effects. The aim of modelling both single and binary lens events is to produce an accurate model that produces the same change in magnification over time, called a light curve, that represents the real data.

2.5.1 Light curves

Any microlensing model can be used to produce a light curve to visualize the change in magnification of an event. A point source single lens requires three parameters (u_0, t_0 and t_E, detailed in Appendix B) and Equation 2.9, to produce a Paczyński curve. Whereas, shown in section 2.4.1 the magnification for a binary lens is more complex, requiring the solution of a 5^{th} order polynomial. Solving for the image positions requires the source position, lens positions and their masses. As with a single lens, the motion

of a source star behind the lens can be traced across the lens plane seen in Figure F.1. The fractional masses (ϵ_i) can be determined from a single variable, q, the mass ratio, where $\epsilon_1 = \frac{1}{1+q}$ and $\epsilon_2 = \frac{q}{1+q}$. Both masses can be defined to be on an axis of the lens plane and their positions described by a single term, d, the lens mass separation. d and q are all that are required to describe the binary lens masses, yet the source position still needs to be described. The source position (u) can be expressed as a two dimensional coordinate u_1 and u_2, which can be expressed as a complex coordinate ($\zeta = u_1 + u_2 i$) when applied to the binary lens equation. The coordinates u_1 and u_2 are described by four physical parameters, the distance of closest approach (u_0), the angle of source trajectory (ϕ), the time of closest approach (t_0), and the Einstein time (t_E), as detailed in Appendix F. Parameters u_0 and ϕ are defined as the closest approach to the centre of mass at an angle anti-clockwise from the positive x-direction. They change the path of the source trajectory and therefore the shape of the light curve, where-as the time parameters (t_0 and t_E) move and scale the light curve in time, an example of which is shown in Figure 2.2.

2.5.2 Magnification maps

A technique commonly used to model binary lens events, is by use of a magnification map. This is a map where every pixel depicts the magnification a source would experience at the pixel's coordinates, shown in Figure 2.3. Using this map it is possible to trace a source path (dependant on u_0 and ϕ), reading off the magnification at every point along the trajectory (interpolation), producing a light curve.

To generate a magnification map the only two binary lens parameters required are d and q. A two dimensional grid around the binary lens can be created where every grid point is located at a two-dimensional coordinate in the lens plane (ζ). Solving the 5^{th} order polynomial at each grid point and summing the magnifications of the real solutions produces a point source magnification map.

Magnification maps are useful tools to visually inspect how the magnification changes with source position. The process involved in making these maps is not complex, however, solving the 5^{th} order polynomial is a computationally intense process and can be very slow. Time taken in making the maps is a significant problem when dealing with high resolution maps. The only limit to the accuracy of a magnification map is the resolution of the grid spacing used to make it. As each magnification value on the map is for a point source, any source position that lies between grid points is an approximation, as it needs to be interpolated.

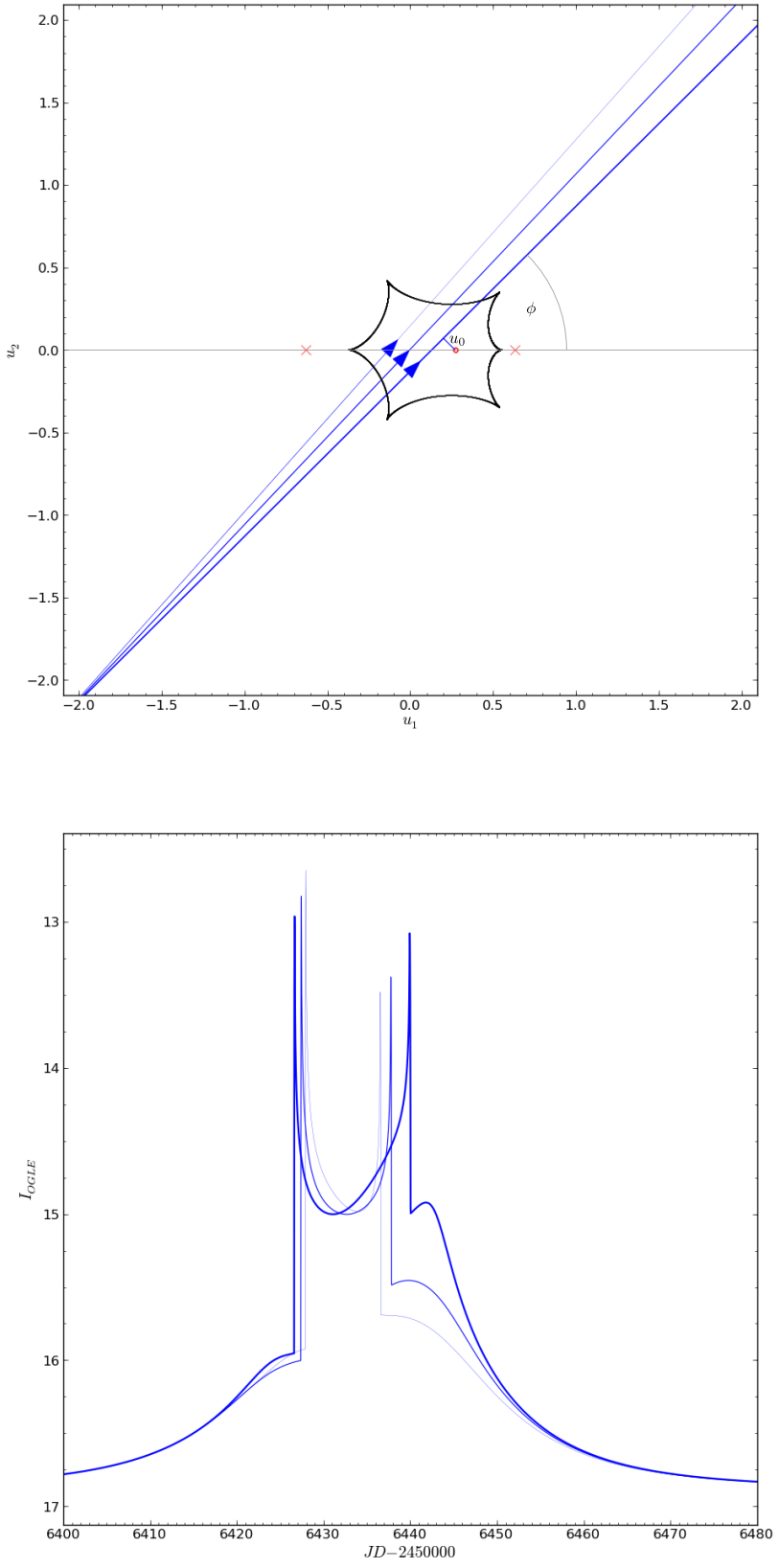


FIGURE 2.2: A map of the lens plane identifying the source trajectory parameters and their effect on the shape of a light curve.

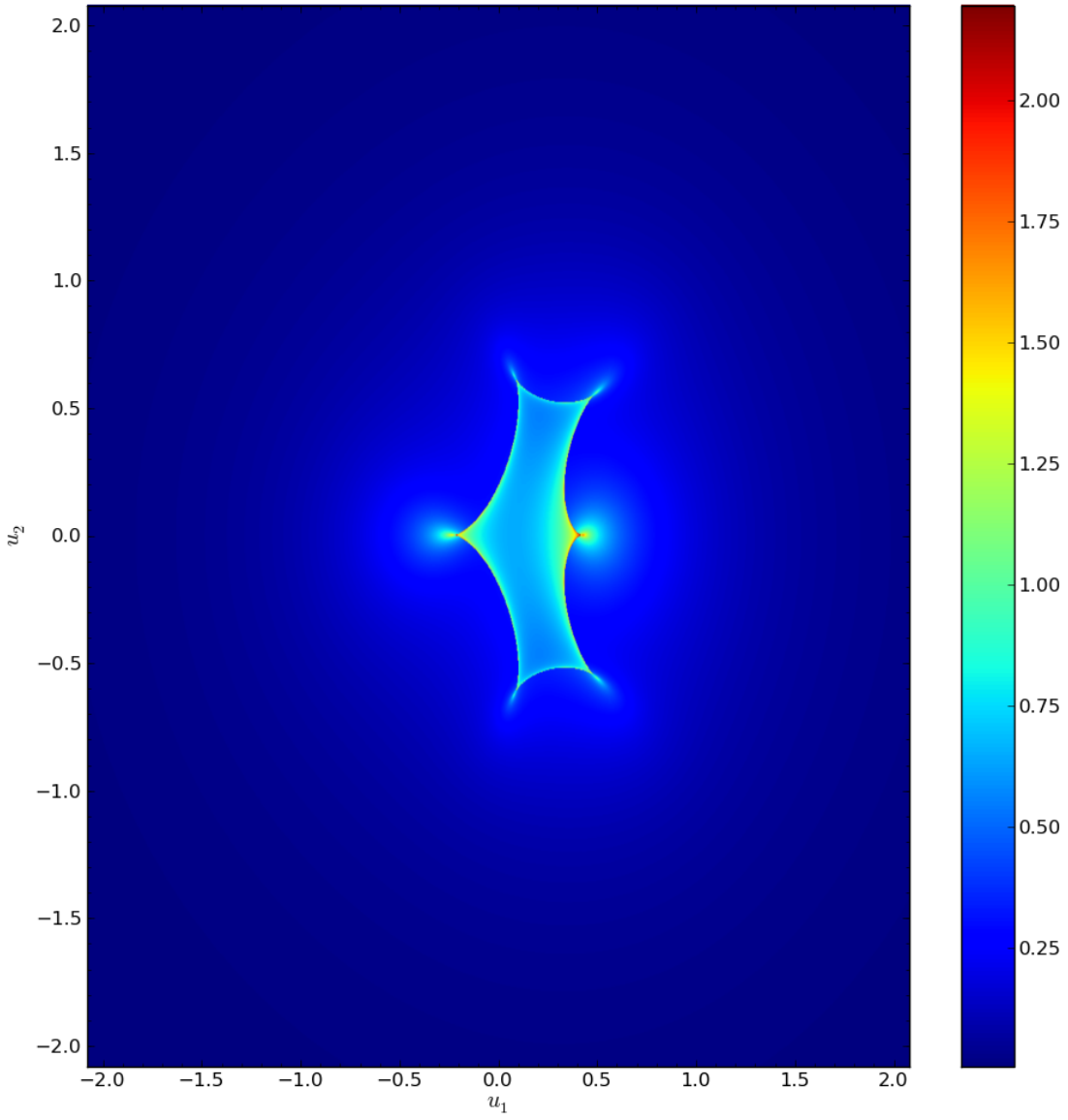


FIGURE 2.3: A magnification map (in the lens plane) showing the change in magnification of the source as a function of position in the lens plane, where the origin is the mid point between two binary lens masses spaced $1R_E$ apart, with a mass ratio of 0.398.

A magnification map enables many different source trajectories to be compared with ease. The light curve of any trajectory across a magnification map can be quickly interpolated without any additional root solving required. Therefore, with a single d, q pair, a magnification map can be made and every possible trajectory can be interpolated.

2.5.2.1 Inverse ray shooting

The production of the magnification map by solving the 5^{th} order polynomial is the most accurate method, but it is computationally expensive. Due to limited computational resources, an alternate method has been developed to produce a magnification map without the need of such complex numerical calculations. This has been achieved by inverse ray shooting (Kayser et al., 1986), which has been used extensively in microlensing modelling research (Schneider & Weiss, 1987; Wambsganss et al., 1990, 1992; Street et al., 2013). The method involves the shooting of rays from the observer back through the lens-plane and determining if they land inside the source. This is achieved by using equation 2.11 and providing image positions (z), then solving for the source positions (ζ). By shooting up to billions of rays and binning them in the source plane, a magnification map can be built up for a given d, q pair. Although solving the inverse ray equation is faster than a 5^{th} order polynomial, it needs to be solved many more times, leading to other potential time issues. A repetitive calculation such as this across a large array is an ideal task for a GPU. In 2010 both Thompson et al. (2010) and Barsdell et al. (2010) developed inverse ray shooting methods using GPUs, with both concluding the method being ideally suited to a GPU and large performance gains were achieved.

2.5.3 Finite source maps

So far the solutions to the lensing equations have only been used to deal with point sources. However, we know that this is not a true representation as the source star will have a physical size. Although it is not possible to resolve the lensing system and measure angular sizes, it is possible to model the size of the source compared with the size of the Einstein radius (θ_E). The introduction of an additional model parameter, ρ, the radius of the source star in units of θ_E, is required to model this effect.

Finite source effects are the main cause of the time consuming calculations required to model microlensing events. For a point source model a single root solving calculation can lead to the magnification of the source. In the case of an extended source a single root solution will not suffice, instead an image area needs to be considered which requires multiple root solutions. Fortunately, finite source effects are relatively easy to include in a magnification map, without the need to solve for them directly. This allows interpolated trajectories to model this effect with minimal additional calculations. To achieve the necessary modifications to a magnification map, a kernel of a finite source must first be produced, before the point source map is convolved (Berry & Burnell, 2006). This technique is fast and allows numerous values of ρ to be convolved with a minimal addition to the computational work.

2.5.3.1 Source star kernel

The kernel is a map that depicts the size of the finite source and how many pixels on a magnification map it encompasses. It is a square matrix where each pixel corresponds to the total brightness of the source it encloses. A source kernel can be used to convolve a magnification map to account for the finite size of the source.

2.5.3.2 Limb darkening

The model used for a finite source has so far assumed the source to be uniformly bright across the whole area of the disk. This is known not to be true as, the centre of the disk would be the brightest point with it getting darker radially outwards. This effect is known as limb-darkening and has been thoroughly studied. Milne (1921) first attempted to describe the effect and produced the linear limb-darkening law (Equation 2.12). Since then further studies have shown the linear law to be inadequate and numerous new limb-darkening laws have been proposed (Klinglesmith & Sobieski, 1970; Manduca et al., 1977; Wade & Rucinski, 1985; Claret & Gimenez, 1990; Diaz-Cordoves & Gimenez, 1992; van Hamme, 1993; Claret, 2000).

For microlensing it has been shown that, although linear limb-darkening is not an ideal model of stellar atmospheres, it is sufficient (Bennett, 2010). The inclusion of the following limb-darkening function has been included when modelling finite source events,

$$I_N = 1 - \Gamma \left(1 - \sqrt{1 - \left(\frac{r}{\rho} \right)^2} \right) \quad , \tag{2.12}$$

where I_N is the normalized intensity at a distance r from the centre of the source of radius ρ, and Γ is the linear limb-darkening coefficient.

The production of a kernel for uniformly bright sources can be calculated by determining the fraction of the source which is inside each pixel. For a limb darkened source, the position on the source disk which is inside the kernel pixel is also significant when calculating the fraction enclosed. Equation 2.12 can therefore be used to return the intensity of each point of the source enclosed by a kernel pixel, instead of a 1 or 0 representing a point inside or outside of the source respectively.

2.5.3.3 Convolution

The source star kernel is used in convolving the point source magnification map to produce a finite source magnification map. A normalized source star kernel of the same resolution as the point source map is required to correctly convolve the map. The convolution process is applied to every pixel of the point source magnification map, it involves summing the multiplication of each kernel pixel with the corresponding point source map's pixels. A graphical representation of this is shown in Figure 2.4.

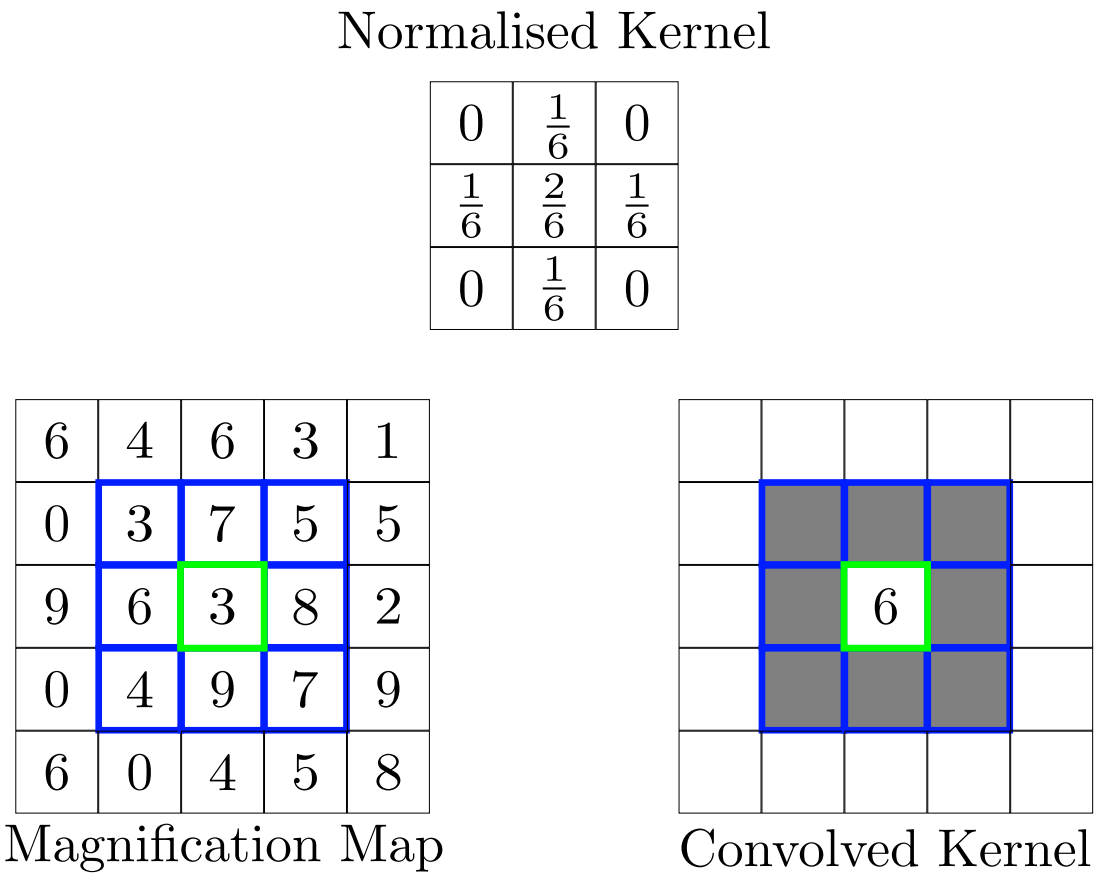


FIGURE 2.4: Showing how convolution works on the central pixel, by summing the product of the kernel and an original source to produce a convolved value.

2.5.4 Alternate modelling approaches

To model a single light curve using the above method, requires making a magnification map to interpolate a single trajectory from. This means numerous numerical calculations are performed for a single interpolation, which doesn't require every pixel's magnification and is time consuming to process. Alternate methods (Dominik, 1993; Gould, 2008; Bennett & Rhie, 1996) and combinations of these methods (Dong et al., 2006) have therefore

been developed to model microlensing events without the need of computationally time consuming magnification maps.

2.5.4.1 Contour integration

A method developed to avoid the need of magnification maps is contour integration. Originally used by Schramm & Kayser (1987), this method was extensively develoepd further in following years by Dominik (1993, 1995, 1996, 1998a, 2007), with additional work performed by Gould & Gaucherel (1997) and more recently a well documented implementation developed by Bozza (2010). The basis of contour integration is to solve the point source lens equation as few times as possible to determine an acceptable approximation of the contours of each image. The starting point of this method is solving the lens equation for a few points on the source contour, the image positions are sorted and then the image contours are traced. By use of error estimators the process determines if more image positions are required. These come from solving the lens equation at additional points on the source contour. The process iteratively builds up a set of points that trace the contour of each image. These contours are used to determine the total area of the images, using a trapezium approximation of Green's line integral, and thus the magnification of the source can be found.

A Matlab implementation of this method has been developed, but optimal performance could not be achieved. A significant portion of time is spent re-ordering the image positions to construct the line integral, which increases in complexity as the source approaches caustic structures.

2.5.4.2 Hexadecapole

Hexadecapole is another alternate method developed to rapidly model extended source microlensing events. Based upon the work of Pejcha & Heyrovský (2009) (initial submission in 2007), Gould (2008) shows that by using a Taylor expansion to describe the source area, it is possible to derive an expression to determine the magnification of a source requiring only 13 point source calculations,

$$A_{finite} = A_0 + \frac{A_2\rho^2}{2}\left(1 - \frac{1}{5}\Gamma\right) + \frac{A_4\rho^4}{3}\left(1 - 1\frac{11}{35}\Gamma\right) + \ldots \ , \tag{2.13}$$

where

$$A_2\rho^2 = \frac{16A_{\rho/2,+} - A_{\rho,+}}{3} \ , \ A_4\rho^2 = \frac{A_{\rho,+} + A_{\rho,\times}}{2} - A_2\rho^2 \ . \tag{2.14}$$

$A_{w,+}$ is the average of point source magnification differences between the source centre and four points at a radius w, equally spaced angularly, $A_{w,\times}$ is the same but with an angular offset of $\frac{\pi}{4}$. These expressions show that the 13 point source calculations required come from 8 equally spaced points around the source at a radius of ρ, 4 at a radius $\frac{\rho}{2}$, and 1 at the centre of the source.

Gould proceeds to derive the error due to the hexadecapole approximation and states it is only sufficient when the source is not too close to caustic structures, in which case alternate modelling methods are required. The hexadecapole method may not be able to model all source positions in a lensing trajectory but, when it is valid, it can produce performance improvements up to several orders of magnitude over direct root solving solutions.

2.5.4.3 Image centered inverse ray shooting

An Image Centered Inverse RAy Shooting (ICIRAS) approach is a method developed by Bennett & Rhie (1996) and further detailed by Bennett (2010). Using this method it is possible to model all seven binary lens parameters, and only minor adjustments are required to include higher order effects, and triple lens modelling.

The modelling process involves locating the centre of each lensed image and setting up a grid to encompass the full image. Inverse ray shooting is performed at each grid point to determine if the ray falls within the source, therefore the grid point is inside the image. The grid can then be integrated in two dimensions to determine the total brightness of the image. Summing each individual image's brightness and dividing by the brightness of the source determines the total magnification. Limb-darkening can easily be incorporated into this method by making the inverse ray shooting function return a limb-darkened intensity for any rays that land inside the source.

The shape of the lensed images are typically curved and follow the edges of the critical curves, which are often long and narrow, potentially even full rings. Hence, polar coordinates are used as the coordinate system to closely match the shape of the images (shown in Figure 2.5). This results in an improved accuracy when integrating, due to a grid in polar coordinates encompassing the whole image without including much of the area outside of the image, resulting in a high density of grid points to be integrated over, which does not often occur when performed in cartesian coordinates. Of the three to five image positions determined a grid is created for each separately, which are integrated in both dimensions and summed with all other images to find the total brightness.

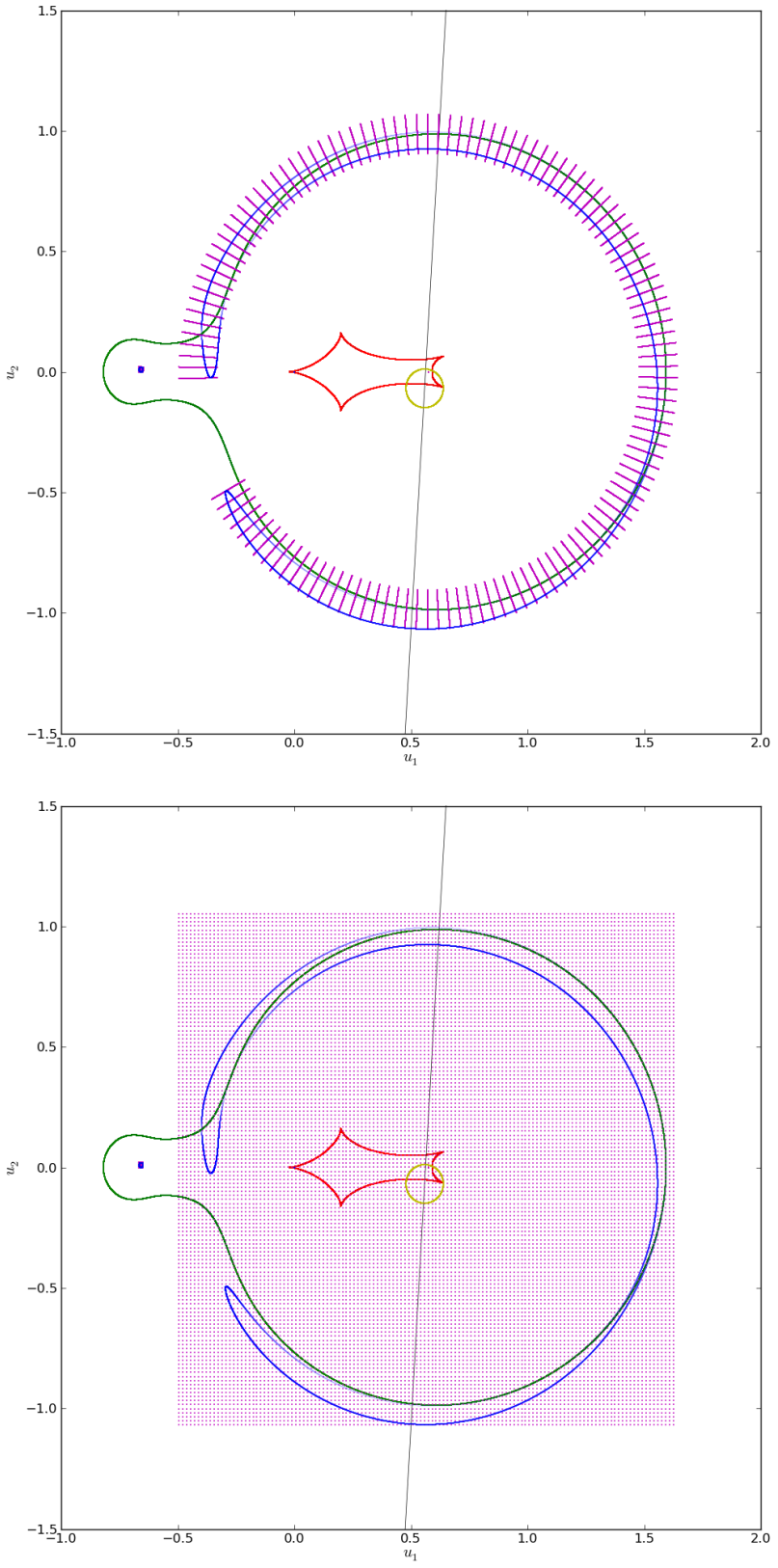


FIGURE 2.5: Image grids (purple), encompassing all images (blue) caused by a finite source (yellow) crossing the caustics (red) of a binary lens. The top figure shows the grid distributed in polar coordinates with the origin at the centre of mass. The bottom figure shows a grid of the same number of points (128x128) in cartesian coordinates encompassing exactly the same images.

The accuracy of the image boundary during the integration is important to avoid numerical errors, so Bennett (2010) introduced a mathematical operation to account for this without the need of excessive oversampling to accurately locate the image boundary,

$$\int_{y_{L1}}^{y_{L2}} f(y)\,\mathrm{d}y = h(A_1 f(y_{L1}) + B_1 f(y_1) + f(y_2) + ... + f(y_{N-1}) + B_N f(y_N) + A_N f(y_{L2})) \ , \tag{2.15}$$

where $L1$ and $L2$ correspond to the closest grid points outside and either side of the image, and A_i and B_i are determined by a Heavyside step function dependent on the resolution of the grid spacing, as detailed by Bennett (2010).

To compare a model to data the magnification of the source needs to be determined, however the ICIRAS method detailed here only calculates the image brightnesses. As shown in section 2.4.1, the magnification of a source is a ratio of the total brightness of the images to the brightness of the source. For a uniform source, the source brightness would be equal to its area. However, with the inclusion of limb-darkening, the brightness function needs to be integrated in polar coordinates to determine its total brightness. For a linear limb-darkened source this produces a total brightness of,

$$\int_0^{2\pi}\int_0^{\rho} 1 - \Gamma\left(1 - \sqrt{1 - \left(\frac{r}{\rho}\right)^2}\right) r\mathrm{d}r\mathrm{d}\theta = 2\pi\rho^2\left(\frac{3-\Gamma}{6}\right) \ . \tag{2.16}$$

2.5.5 Higher-order effects

The microlensing of a source star has so far been modeled by a point or disk moving across the lens plane in a straight line, but this does not tell the whole story. Other known effects exist that have not been accounted for in the model so far, such as the Earth's motion, the multiple lens system's orbital motion, and photometric variability in the source star. These effects cause small perturbations in the light curve and can be included in the modelling to learn more about the microlensing system, but to include these effects requires additional parameters. The perturbations caused by these effects are typically subtle, and are often only determined if the event is observed with a high cadence and small photometric errors.

2.5.5.1 Parallax

Parallax is the apparent movement of an object caused by the change in position of the observer. Observations of microlensing events are made throughout the year, during which time the Earth orbits the Sun, and so the observations are taken from different positions. This results in an apparent motion of the source across the lens plane, which is not correctly modeled by a straight path traveling at a constant rate. Instead, two additional parameters are required to describe this deviation in source trajectory. Gould (2004) describes a geocentric coordinate system where the inclusion of parallax has minimal effect on the remaining parameter values. The source position $u(t)$ is made up of two components,

$$u(t) = \sqrt{\tau(t)^2 + \beta(t)^2} \quad , \tag{2.17}$$

where τ is the projection along the direction of motion, and β is the perpendicular separation from it. Without parallax, $\tau = \frac{t-t_0}{t_E}$ and $\beta = u_0$, which produces the expected non-parallax image position u shown in Equation 2.10. With the inclusion of parallax, perturbations in both τ and β are introduced,

$$\tau(t) = \frac{t - t_0}{t_E} + \partial\tau(t) \quad , \quad \beta(t) = u_0 + \partial\beta(t) \quad . \tag{2.18}$$

Relative to non-parallax (rectilinear) motion, the source position is,

$$(\partial\tau(t), \partial\beta(t)) = \pi_{\mathbf{E}}\mathbf{\Delta s} = (\pi_{\mathbf{E}} \cdot \mathbf{\Delta s}, \pi_{\mathbf{E}} \times \mathbf{\Delta s}) \quad , \tag{2.19}$$

where $\mathbf{\Delta s}$ is the apparent position of the sun relative to where it would be if rectilinear motion of the Earth is assumed, and $(\pi_{\mathbf{E}})$ is the two parameter parallax vector. Using a geocentric time frame, it is possible to determine $\mathbf{\Delta s}$,

$$\mathbf{\Delta s}(t) = \mathbf{s} - (t - t_p)\mathbf{v_p} - \mathbf{s}(t_p) \quad , \tag{2.20}$$

where $\mathbf{s}$ is the Earth-Sun vector, $\mathbf{v_p}$ is the velocity vector of the Earth at the point, and t_p is a fixed time. The geometry of the movement and parallax terms can be seen in Figure 2.6. Therefore, with the additional knowledge of the Sun's position $(\mathbf{\Delta s})$ and the inclusion of the parallax vector $(\pi_{\mathbf{E}})$ it is possible to describe the apparent motion of the source across the lens plane caused by the movement of the Earth.

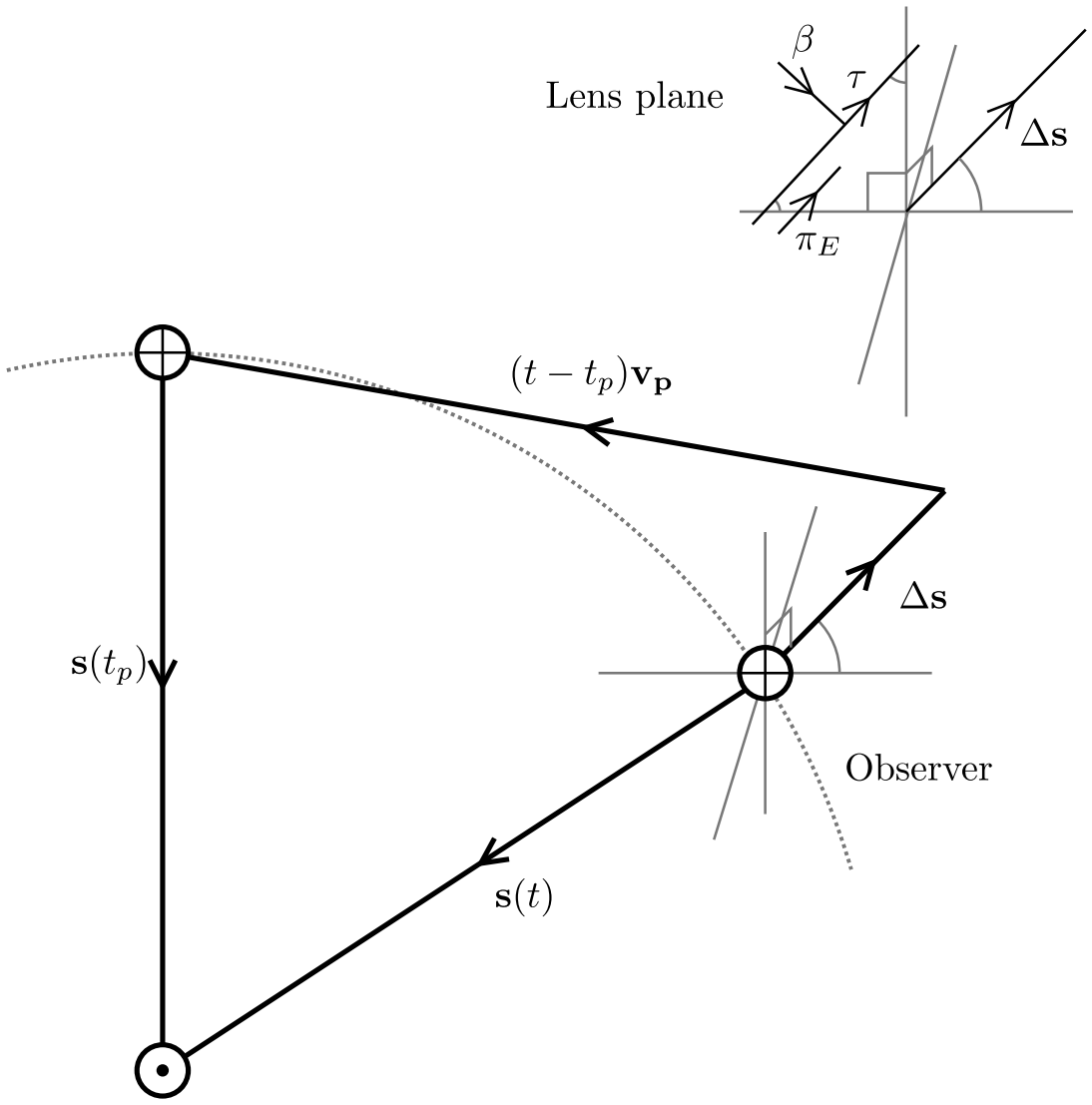


FIGURE 2.6: Geometric representation of the parallax vector and Earth's motion, showing the relation between the Earth-Sun vector and the resulting parallax vector.

2.5.5.2 Orbital Motion

Not only does the Earth orbit around the Sun, but a multiple lens system will also have orbital motion. In reality this should be described by a full Keplerian orbital parameter set (5 additional parameters) (Dominik, 1998b). However, under the assumption that only a small part of the lens orbital trajectory is covered during a microlensing event, a simpler two parameter description can be used (Albrow et al., 2000; Shin et al., 2011; Skowron et al., 2011). The two-parameter binary lens orbital motion approximation makes the assumptions that the binary lens rotation and separation change at a constant rate, introducing two new parameters. $\frac{\partial d}{\partial t}$, the rate of change in lens object separation, and $\frac{\partial \phi}{\partial t}$, the rate of change in angle between the lens axis and source trajectory. These

two parameters can be incorporated into the model by addition with the appropriate terms, $d = d(t_0) + \frac{\partial d}{\partial t}(t - t_0)$ and $\phi = \phi(t_0) + \frac{\partial \phi}{\partial t}(t - t_0)$.

2.5.5.3 Binary source

An additional consideration that causes perturbations in the light curve occurs when the source is a binary star system, which can result in several effects. When the binary source stars are widely separated, they can be modeled as two separate sources following the same or similar trajectories. If their orbital motion is taken into account, their trajectories across the lens plane may no longer be a simple straight line. Should the binary source stars be eclipsing, it leads to variations in the source brightness which also need to be modeled.

2.5.5.4 Xallarap

As mentioned previously, if the source object is a binary system, the source star(s) will have orbital motion. Just as the Earth orbits around the centre of mass and causes deviations from the source's rectilinear trajectory across the lens plane (Parallax), the binary source will be orbiting its centre of mass, and likewise will cause deviations from the source's rectilinear trajectory (Griest & Hu, 1992; Han & Gould, 1997). This affect is analogous to parallax but in reverse, hence the name (Bennett, 1998). Given that Xallarap is analoguos to Parallax, it comes as no surprise that it too can be modelled by two parameters which describe the orbital rotation of the binary source. This method is used to try and detect exoplanets, not around the binary lens but instead, around the binary source (Rahvar & Dominik, 2008, 2009).

2.5.5.5 Others

In addition to the higher-order effects mentioned above, several others have been considered and statistically modeled to determine their significance. One such method detailed by Heyrovský & Sasselov (2000) and Rattenbury et al. (2002) is the effect of stellar spots on the source star, however, Rattenbury goes on to conclude that this affect is likely to have negligible effect on the light curve.

Other research has investigated the detectability of extrasolar moons (Liebig & Wambsganss, 2010). Liebig and Wambsganss not only show that it is possible to detect an extrasolar moons using a triple lens model, but also propose that it is "within close reach

of available observing technologies", stating that "under very favourable conditions, exomoons may already be observable".

An alternate cause of perturbations in the light curve which has already been hinted at, comes from another special situation of the binary source. It consists of a planetary companion which transits the stellar source (Rybicki & Wyrzykowski, 2014). Rybicki and Wyrzykowski study the effects of a transiting planet around a source star being microlensed by a single lens. They show the possibility of such an occurrence, the problems with degeneracies of eclipsing binary sources, and finally conclude that with current microlensing observational surveys the chances of observing such a phenomenon is very small (2×10^{-6}).

2.6 Summary

This chapter presents the origins of microlensing and a detailed description of what it is. Starting with a brief history of the work performed by Bohdan Paczyński (1986b), an overview of the different microlensing research groups past and present are shown, followed by the mathematical derivation of the lensing equations. Detailed descriptions of both single and binary lens equations are presented, including the determination of an event's magnification from its lensed image positions.

A description of how microlensing events are modeled is shown. With the inclusion of finite sources, the modeling of an event becomes more complex, and the multiple ways that attempt to overcome this challenge are presented, with the strengths and weaknesses of the methods being highlighted. Additional higher-order effects are also detailed, explaining their cause and what effect they have on an event.

Chapter 3

Data Sources

3.1 Observatories

There are a number of different microlensing research projects in existence, with the majority of data coming from the two current survey telescopes which are part of the OGLE and Microlensing Observations in Astrophysics (MOA) collaborations.

The OGLE[1] project began observing in 1992 on the 1m Swope telescope at the Las Campanas Observatory in Chile, operated by the Carnegie Institution of Washington. They observed for a total 65 nights using a single 2048×2048 Loral CCD detector which gave a resolution of 0.44" per pixel, covering a total area of $15' \times 15'$ in a single frame (Udalski et al., 1992). The project's success lead to development of an Early Warning System (EWS) in 1994 (Udalski et al., 1994), and continual improvements in the technologies used including the EWS (Udalski, 2003). Currently the OGLE project is operating in its fourth phase, using a new mosaic camera, with 2048×4102 pixel CCD detectors, each with a resolution of $0.26''$, providing a total area of 1.4 square degrees per exposure (Soszyński et al., 2012).

The MOA[2] project is a collaboration between New Zealand and Japan and was announced at the 12^{th} IAP Astrophysics meeting in 1996 (Abe et al., 1996). Originally they used the 0.6m Boller and Chivens telescope with a nine CCD mosaic camera attached, where each CCD had 1024×1024 pixels, providing a $0.53° \times 0.53°$ field of view at the Mt. John University Observatory, New Zealand. In 2002 a grant was provided for the construction of the currently operated 1.8m MOA telescope at Mt. John University Observatory, New Zealand, with a new camera (MOA-cam3), receiving first light on the 15^{th} of April 2005 (Hearnshaw et al., 2006). MOA-cam3 was designed specifically for use

[1] OGLE - http://ogle.astrouw.edu.pl/ on 2014 June 04
[2] MOA - http://www.phys.canterbury.ac.nz/moa/ on 2014 June 04

with the MOA project and consists of ten (5×2) CCD detectors each with 2048×4096 pixels (providing a total area of 10240×8192) and is cryogenically cooled to around $-80°C$, when installed each pixel has a field of view of $0.58''$ providing a total coverage of $1.32° \times 1.65°$ per exposure (Sako et al., 2008).

Although there are many other sources of data, these two telescopes are dedicated to microlensing surveys towards the centre of the Galaxy, Large Magellanic Cloud (LMC) and Small Magellanic Cloud (SMC), returning thousands of microlensing detections (2220 unique microlensing detections in 2013). Large detection rates are obtained due to the length of time dedicated to this research, the telescopes being designed specifically to detect this phenomenon, and with the instruments being upgraded over time to enhance their scientific output. The remaining microlensing data comes from professional follow up groups such as Probing Lensing Anomalies NETwork (PLANET) and RoboNet, with additional data from the MicroFUN group which has a large contingent of amateur astronomers.

Founded in 1995 the PLANET[3] collaboration's primary goal is to study anomalies in microlensing events by performing high cadence observations with continual coverage of anomalous microlensing events (Albrow et al., 1996, 1998). Four separate observing sites were part of the collaboration which provided nearly continuous monitoring of events. One of these sites was located at La Silla, Chile, where the ESO Dutch 0.92m telescope was used, it was fitted with a 512×512 CCD camera with each pixel having a resolution of $0.44''$. Another observing site was located at Sutherland, South Africa, where the SAAO 1m Elizabeth telescope used a 512×512 CCD with each pixel having a resolution of $0.35''$. Additionally an observing site at Bickley, Australia, was included which hosted the Perth 0.6m telescope fitted with a 576×384 CCD where each pixel had a resolution of $0.58''$. Finally a site was located in Hobart, Australia, where the University of Tasmania 1m telescope is housed, it was fitted with a 512×512 CCD with each pixel having a resolution of $0.47''$ (Dominik, 1996). Initial work began in 1995 where they performed follow up observations of nine Galactic bulge events. Since their first year of operation the PLANET collaboration have observed around 1000 unique events aided by access to additional telescopes. In 2005 they joined together with the RoboNet-1.0 project to improve global coverage, which was further improved in 2009 when they also joined with the MicroFun group.

The RoboNet[4] project uses an automated target prioritizing system to perform high cadence follow-up observations on a network of robotic telescopes. Originally started in 2004 (Bode et al., 2004), the research strategies were very similar to those of PLANET,

[3]PLANET - http://planet.iap.fr/ on 2014 June 04
[4]LCOGT - http://lcogt.net/ on 2014 June 04

hence the earlier mentioned collaboration, but with a different primary goal being the detection of cool extra-solar planets. RoboNet uses the resources of the Las Cumbres Observatory Global Telescope Network (LCOGT) to achieve a worldwide network enabling continual coverage of any microlensing event (Tsapras et al., 2009). LCOGT consists of a 2m telescope located at Haleakala, Hawaii, a 2m and two 1m telescopes at Siding Spring, Australia, three more 1m telescopes are located at the Las Cumbres Observatory, Chile, another three at Sutherland, South Africa, and one more at the McDonald Observatory, Texas, USA. Each 2m telescope is initially fitted with an Merope LLC CCD camera which has 2048×2048 pixels and a field of view of $4.7' \times 4.7'$, while each 1m telescope is fitted with an SBIG STX-16803 consisting of 4096×4096 pixels for a field of view of $16' \times 16'$. Continual improvements are being made to the network with newer cameras and additional telescope of 0.4m and 1m already planned for, or in the process of being installed for use in the near future.

Forty six different observational set ups have joined the MicroFUN[5] consortium, they range from 0.25m to 2.4m telescopes with a variety of photmetric instruments (a full detailed list can be found on the groups website). They are an informal organization spread across five continents comprised of professional and amateur astronomers, with the shared goal of performing follow up observations of microlensing events. With data archives going back to 2003, they are an active group providing crucial follow up data of high magnification events.

The follow up groups are alerted by the survey teams of ongoing events with high magnification. The magnitude of an event is often a limiting factor to follow up teams, as they often have smaller equipment and are not able to observe objects as faint as the large survey telescopes. Equipment limitations mean that usually only the very bright, or highly magnified events receive additional follow up measurements.

3.2 Multi site data

To achieve the most accurate models, as much data as possible should be considered. This requires data from multiple sources located all over the globe, ideally providing continual coverage throughout a whole event (i.e. 24 hours a day). There is no convention for all observational data to be made available in the same way and therefore data coming from different sites can be in different forms. Data received from telescopes are usually either in magnitudes, m_i, or differential flux, ΔF_i which are related by,

[5]MicroFUN - http://www.astronomy.ohio-state.edu/ microfun/ on 2014 June 04

$$m_i = m_c - 2.5 \log (F_0 + \Delta F_i) \quad , \tag{3.1}$$

where m_c is the calibration magnitude and F_0 is the flux of the event measured from a reference image (ideally baseline). For simplicity it is chosen to convert all magnitude data into differential flux measurements, adopting $F_0 = 0$ so that

$$\Delta F_i = 10^{-0.4(m_i - m_c)} \quad , \tag{3.2}$$

with uncertainties of (determined by $\sigma_x = \frac{\mathrm{d}x}{\mathrm{d}y}\sigma_y$),

$$\sigma_{\Delta F_i} = -0.4 \ln(10) e^{-0.4 \ln(10)(m_i - m_c)} \sigma_{m_i} \quad . \tag{3.3}$$

The two main sources of microlensing data are from the MOA and OGLE collaborations, whose telescopes perform wide surveys nightly producing a large quantity of data for multiple events. For MOA data, all measurements are provided in units of delta flux (ΔF). OGLE data, and most other observations coming from follow-up telescopes are provided in magnitudes.

The observed total flux (F_i) of a microlensing event is comprised of two components, described by,

$$F_i = A_i F_s + F_b \quad , \tag{3.4}$$

where A_i is the magnification, F_s is the source flux, and F_b is the baseline flux, and

$$\Delta F_i = F_i - F_0 \quad . \tag{3.5}$$

If the reference (F_0) was taken exactly at baseline it would be equal to $F_s + F_b$, however this cannot be assumed, resulting in,

$$\Delta F_i = F_s A_i + F_b - F_0 \quad . \tag{3.6}$$

Which can be written as,

$$D_i = a_0 A_i + a_1 \quad , \tag{3.7}$$

where D_i is the recorded data (ΔF_i), a_0 is the first coefficient (F_s), and a_1 is a constant ($F_b - F_0$). To determine the best solutions of a_0 and a_1 for a given magnification model $A_i = A(t_i)$, linear regression is used. This requires an expression to describe their goodness of fit,

$$\chi^2 = \sum_{i=1}^{N} \frac{(a_0 A_i + a_1 - D_i)^2}{\sigma_{D_i}^2} \ . \tag{3.8}$$

As with any χ^2 function, the minimum is equal to the best fitting solution, therefore by differentiating the equation with respect to the unknowns they can be equated to zero to determine the minimum,

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}a_0} = 2\sum_{i=1}^{N} \frac{A_i(a_0 A_i + a_1 - D_i)}{\sigma_{D_i}^2} = 0 \ , \tag{3.9}$$

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}a_1} = 2\sum_{i=1}^{N} \frac{a_0 A_i + a_1 - D_i}{\sigma_{D_i}^2} = 0 \ . \tag{3.10}$$

From which,

$$\sum_{i=1}^{N} \frac{D_i A_i}{a_0} = a_0 \sum_{i=1}^{N} \frac{A_i^2}{\sigma_{D_i}^2} + a_1 \sum_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} \ , \tag{3.11}$$

$$\sum_{i=1}^{N} \frac{D_i}{a_1} = a_0 \sum_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} + a_1 \sum_{i=1}^{N} \frac{1}{\sigma_{D_i}^2} \ . \tag{3.12}$$

These expressions can then be described by its equivalent matrix algebraic form,

$$\begin{bmatrix} \sum\limits_{i=1}^{N} \frac{A_i^2}{\sigma_{D_i}^2} & \sum\limits_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} \\ \sum\limits_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} & \sum\limits_{i=1}^{N} \frac{1}{\sigma_{D_i}^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum\limits_{i=1}^{N} \frac{D_i A_i}{\sigma_{D_i}^2} \\ \sum\limits_{i=1}^{N} \frac{D_i}{\sigma_{D_i}^2} \end{bmatrix} \ . \tag{3.13}$$

which can be inverted and solved to obtain,

$$a_0 = \frac{\sum\limits_{i=1}^{N} \frac{D_i A_i}{\sigma_{D_i}^2} \sum\limits_{i=1}^{N} \frac{1}{\sigma_{D_i}^2} - \sum\limits_{i=1}^{N} \frac{D_i}{\sigma_{D_i}^2} \sum\limits_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2}}{\sum\limits_{i=1}^{N} \frac{A_i^2}{\sigma_{D_i}^2} \sum\limits_{i=1}^{N} \frac{1}{\sigma_{D_i}^2} + \left(\sum\limits_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} \right)^2} \ , \tag{3.14}$$

$$a_1 = \frac{-\sum_{i=1}^{N} \frac{D_i A_i}{\sigma_{D_i}^2} \sum_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} + \sum_{i=1}^{N} \frac{D_i}{\sigma_{D_i}^2} \sum_{i=1}^{N} \frac{A_i^2}{\sigma_{D_i}^2}}{\sum_{i=1}^{N} \frac{A_i^2}{\sigma_{D_i}^2} \sum_{i=1}^{N} \frac{1}{\sigma_{D_i}^2} + \left(\sum_{i=1}^{N} \frac{A_i}{\sigma_{D_i}^2} \right)^2} \quad . \tag{3.15}$$

This process works well for data that comes as ΔF, such as from MOA, however a subtly different approach is required for data that is provided in magnitudes, such as from OGLE.

From equation 3.2, the delta flux can be expressed in magnifications using 3.6 (where $F_0 = 0$),

$$F_s A_i + F_b = 10^{-0.4(m_i - m_c)} \quad , \tag{3.16}$$

where, m_c is taken to be equal to the baseline magnitude provided via the alerted event's webpage (the OGLE EWS baseline magnitude is used if both MOA and OGLE data exists). To express this equation in simpler terms, the following substitutions are made, $D_i = 10^{-0.4(m_i - m_c)}$, $a_0 = F_s$ and $a_1 = F_b$, which produces,

$$D_i = a_0 A_i + a_1 \quad , \tag{3.17}$$

permitting the same minimization procedure to be adopted to solve for the two parameters a_0 and a_1. Converting all data to the same type simplifies the modelling process by allowing a generic modelling method to be used.

3.3 Summary

A range of observing groups contribute to the data of microlensing events. The main contributors are presented in this chapter, briefly detailing their history and the research equipment used. As the data retrieved from each observing group is not in the same format, different treatments of the data are required. Starting from the initial data retrieval a detailed description and justification of the data conversion from different sources into a common form is presented.

Chapter 4

Graphics Processing Units

4.1 Outline

A Graphics Processing Unit (GPU) is a common component found in most desktop computers, but only recently has the potential of these devices for general computation been realized, in a range of disciplines from sciences to finance (Gaikwad & Toke, 2009; Sainio, 2010; Isborn et al., 2011; Mashimo et al., 2013). By utilizing the large scale parallelization ability of these devices, increased performance can be achieved for some computational calculations if correctly incorporated.

4.2 Advantages of a Graphics Processing Unit

Computer technologies have been improving at a tremendous rate over the last few decades and the performance of an average personal computer is far superior to its recent predecessors. Despite these technological improvements, scientific and mathematic computational challenges are sometimes still left unsolvable within a lifetime, often due to the overwhelmingly large quantity of data that is unable to be processed fast enough. Unfortunately these advances in computer performance are slowing down due to the physical limits of the techniques involved in developing the Central Processor Units (CPUs). However, alternate methods such as hyper threading provide new ways that a CPU can be enhanced, improving the performance without the need of increasing the clock cycles of the processors. Another common technique to improve performance is to include more than one core in a processor; thus making a parallel device. Parallel devices are becoming the norm and it is difficult to purchase a new CPU which is not a dual-core, quad-core or even larger multi-core. By having N cores in a CPU, there is the potential to do up to N times the work in the same time. It is instantly apparent that

great gains can be made if you have hundreds or thousands of cores. Such computers exist, which have large numbers of parallel devices carefully set up to produce huge peak performances. Computational performance can be measured in floating-point operations per second (FLOPS). As of the end of 2013 an average laptop can achieve approximately 4 GFLOPS, and a top-end desktop CPU can achieve a little under 200 GFLOPS. The most powerful super computer in the world in November 2013, has a theoretical peak performance of 54,902,400 GFLOPS (Strohmaier, 2013). It is clear parallelism can produce huge performance gains, but unfortunately these come at large costs, with high performance super computers costing hundreds of millions of dollars to purchase and nearly the same again to run for several years. These machines are usually only owned by national governments, defense companies and other very large organizations.

With the power of supercomputers out of reach to smaller groups, how can large performance gains from parallelisation be achieved? A GPU, as the name suggests is a specially designed device for producing graphics and visualization. This is achieved by putting many small cores into a single device, which can work in a large parallel way. These cores do not have very good individual performance power in comparison with a modern CPU, but the large number of cores means the total potential peak performance is much larger than the top end CPUs available. In recent years the large computing potential of these devices has been recognized and a lot of research has gone into this field, leading to the development of GPU cards specifically designed for numerical calculation performance rather than graphics generation. The architecture of these cards is such that if a routine can be made to run in parallel, then there is a large potential for significant performance gains, allowing the calculations to compute in a fraction of the time it would conventionally take on a CPU in a serial way. Such gains have already been found to help advance research in numerous fields, such as fluid dynamics (Nagatake & Kunugi, 2010), weather prediction (Michalakes & Vachharajani, 2008), molecular dynamics (Anderson et al., 2008), medicine (Heinrich et al., 2014), and astronomy (Harris et al., 2008; Barsdell et al., 2010).

GPUs are clearly helping make huge steps forwards in fields of science, allowing the analysis of previously inconceivable quantities of data. There have already been studies specifically in the use of GPUs for gravitational microlensing and comparisons made against current CPU modelling implementations (Bate et al., 2010; Thompson et al., 2010), which show significant performance gains. For these reasons the performance of a GPU card is utilized in this research. Compute Unified Device Architecture (CUDA) enabled devices developed by NVIDIA are used exclusively in this research, and their associated CUDA computing framework specific to the devices has been adopted. Each CUDA device has a "compute capability" which describes its limitations and resource availability. The majority of the work detailed in this thesis has used a NVIDIA C2075

card with compute capability 2.0 (a detailed list of the NVIDIA Tesla C2075 capabilities can be found in Appendix H).

4.3 Architecture

CUDA GPU architecture is designed as a grid containing multiple blocks, where each block contains multiple threads as shown in Figure 4.1. Blocks and threads can be arranged to be in up to three dimensions (since compute capability 2.0). All blocks and threads on the GPU have a unique ID allowing them to be easily controlled. However, depending upon the CUDA device used there are limits as to how they are allowed to be distributed across each dimension. The processors (known as cores) on a CUDA GPU device are grouped into multiprocessors (number of cores depends on CUDA architecture version) that share some resources. Each core can start a single thread, and threads are grouped into warps (32 threads), which must be run synchronously. The code that is called by the CPU to run on a CUDA GPU is called a kernel, and it has to be compiled by a special compiler. Thread divergence must always be carefully handled when developing GPU code, but especially so within a warp. Should it occur, there is a large amount of processor idle time, as every thread in the warp will have to work through every diverged path in series, being either turned 'on' or 'off' depending if it is a path that is relevant to its own ID. GPU coding requires an understanding of the device architecture and a thoughtful approach when designing how best to make a process parallel, to ensure it remains within the physical limits of the GPU card used.

4.3.1 Memory types

A CUDA GPU contains its own on-board memory so that it does not have to perform numerous memory transfers to and from the mainboard's memory, which would be time consuming. The on-board memory is split into several types (Figure 4.2), each designed for a different purpose so that large performance gains can be achieved when used appropriately. The design of these different memory types are vital in providing an easy-to-use memory space that helps to avoid numerous memory access issues, such as bank conflicts and race conditions (parallel memory read and write problems).

4.3.1.1 Global memory

Global memory is the main memory type on a CUDA GPU (device memory) where all input and output variables to and from the mainboard are stored, they last the whole

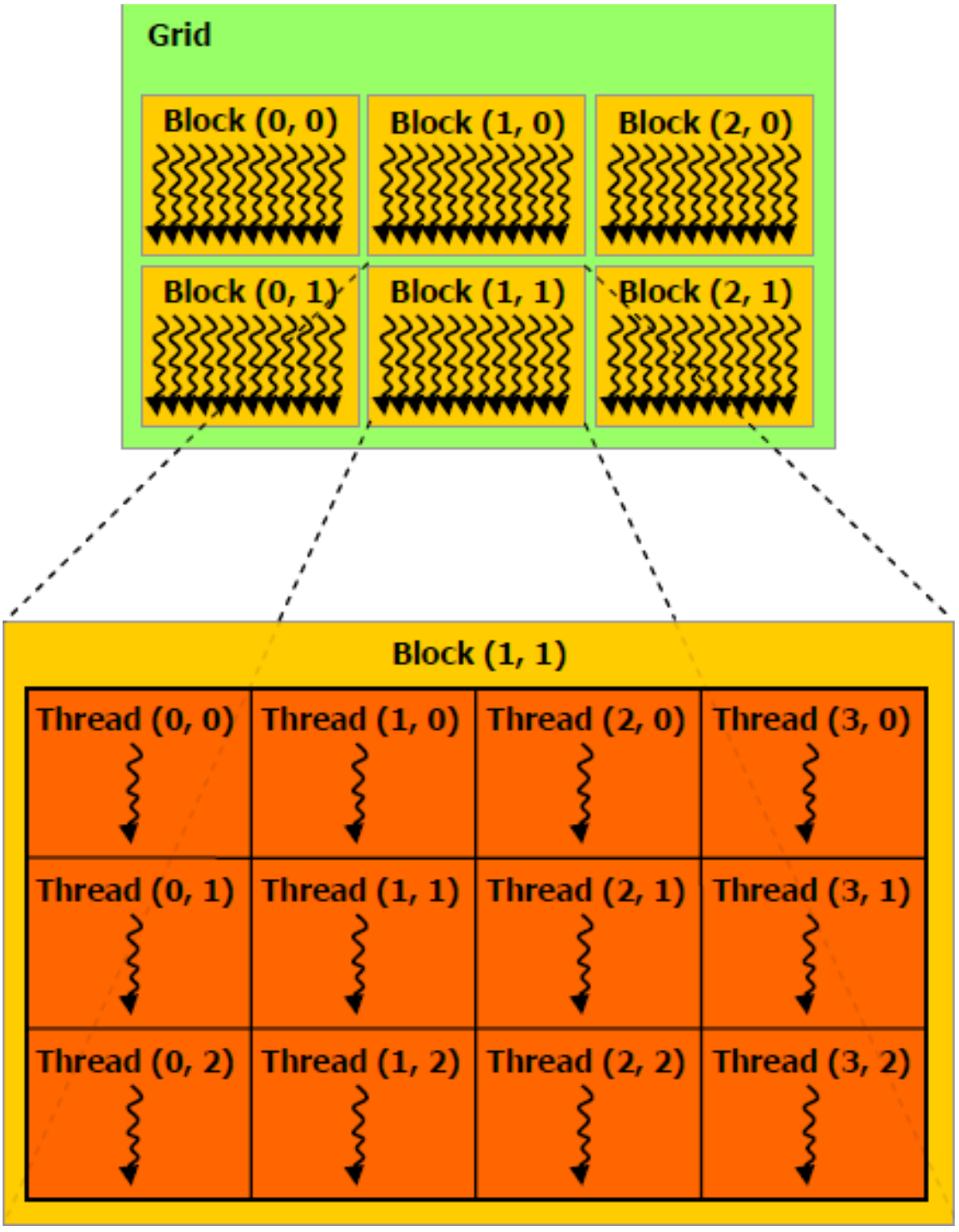


FIGURE 4.1: A pictorial representation of how the threads per block in a grid on the GPU are linked (NVIDIA, 2012).

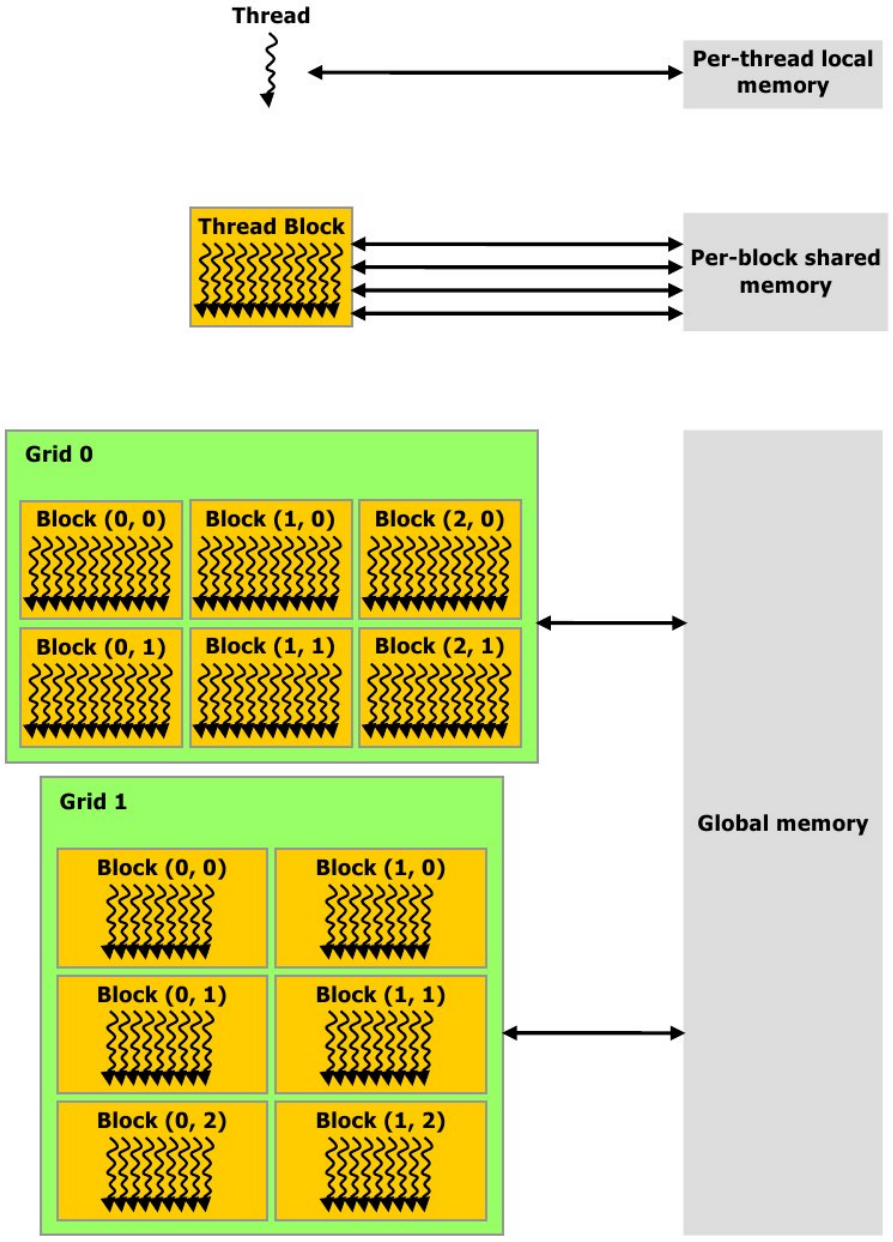


FIGURE 4.2: A graphical view of the available on board memory levels on a CUDA device (NVIDIA, 2012).

length of a kernel's execution. It is not the fastest memory type available to the GPU but it has the largest capacity and is readable and writable by all threads, and all blocks. Due to global memory being accessible by all threads, it is possible that multiple simultaneous memory accesses can result in bank conflicts, hindering performance. Therefore its use needs to be carefully implemented.

4.3.1.2 Shared memory

Shared memory, as the name suggests, is an area of memory that is shared among all threads within a block and persists as long as the block does. It also has the benefit of being the fastest memory type to access, due to it being located on the processing chip. This is the most desired memory space due to the low latency and shared nature, however, it is limited by its maximum size. For compute capability 2.0 there is a very small memory limit of 48KB per multiprocessor, meaning careful management of data is required to fully utilize the limited size of this valued memory type.

4.3.1.3 Local memory

Similar to global memory it is stored in device memory and therefore has the same latency for memory access. It is again readable and writable, however the significant difference is that the memory is only available to individual threads. This allows for every thread to define a unique set of variables without the worry of bank conflicts or race conditions. Each thread is allowed up to 512KB of space, lasting the length of the kernel launch. Local memory is only used by the compiler when it decides that a particular variable is inappropriate or does not fit on the multiprocessor's registry, which has a limit of 32,000 32-bit numbers on compute capability 2.0 cards.

4.3.1.4 Texture memory

Although texture memory is also stored in the device memory (same as global and local memories), it is different in a number of very significant ways. Like global memory it persists as long as the kernel's execution, but is read only. The significant difference with texture memory is how it deals with memory accesses. By using a texture cache it is possible to retrieve multiple memory elements with a single memory read when accessed in a spatially local way, therefore reducing latency.

As the texture memory is cached on a GPU chip, it has faster access times compared with the off-chip GPU DRAM. It was originally designed for graphical applications but

its special traits can also be used for general computation. To achieve the full potential of texture memory it should be used when a code performs spatially-local patterned memory access. This is where code is required to read multiple array values (that are near to each other) in a particular pattern, which is often performed as a repeated task on different areas of the texture map.

The one significant limiting factor when using texture memory is its floating point accuracy. With current CUDA architecture, only single precision floating-point numbers are able to be stored in the texture memory. This could lead to inaccuracies in very high-precision calculations, and should be carefully considered before application.

Hardware-accelerated linear interpolation is an important feature of texture memory. It allows for calls to the texture memory to perform rapid linear interpolations between adjacent element locations in an array on the fly without the need to perform any additional calculation in the users code. This achieves noticeable time gains if interpolation is required, however it is only possible to do linear interpolation, meaning it is not always suitable in every situation.

Additional features of the texture memory deal specifically with boundary conditions, meaning that should a location outside of the stored map's dimensions be requested, it does not result in a segmentation fault. Instead, it is possible to chose one of a few responses, such as always returning zeros, making the requested location coordinates wrap around in a mirrored or un-mirrored way, or returning the nearest array value to the location requested. This feature is particularly useful when applying transformation to a whole array as it does not necessarily need special conditions to deal with the boundaries.

4.4 Optimized coding processes

In the development of software an important process is to ensure the code will run as quickly as possible. For the case of a modelling system that will be analyzing large data sets (which are continually updating) or for trying to model time critical events, performance is a key priority. In code development there exist certain methods to optimize processes where possible, some applicable to all situations and some only applicable to the use of a GPU and its specially developed architecture.

4.4.1 Loops

The process of repeating a set of commands is a common occurrence in many programs. Depending on what is being computed, significant performance gains may be achieved by a process called loop unrolling. As the name suggests it involves 'unrolling' the loop. Instead of having a short loop repeating a very similar step, it can be written as a block of code that performs the loop iterations in a single step. This process cannot be applied to every loop and is only possible for a loop of known size. For example

```
for(i=0; i<5; i++)
{
        value += array[i];
}

// is equivalent to the loop unrolled:

value = array[0] + array[1] + array[2] + array[3] + array[4];
```

4.4.2 Summations

The summation of a large list of numbers can potentially be a time consuming process as conventionally it is performed in series. A single thread takes each element individually and adds them all together. In a parallel environment this is a very wasteful process as all other processors have to wait for a single processor to finish the summation before they can use the output. To avoid this idle processor time occurring and improve the performance of summations, an optimized parallel summation technique has been developed in CUDA (Harris, 2007). Using the loop unrolling method described above it is possible to perform a parallel summation very rapidly. The code to achieve this is detailed below and shows unrolled loops performing summations of a single array. Taking the sum of each element at the thread's ID location (in a 1D block) with the element that is larger by the number of threads in the block, and repeating this process with half the number of threads each time, allows the whole array to be summed rapidly using multiple cores. There is a lot of time to be gained by using unrolled loops, as the number of threads actively computing values instantly decreases by a half each iteration. If this were within a for loop, numerous idle threads would continue looping over or just waiting at the end for the rest to finish. With unrolled loops a special case arrises for the last 32 threads. Due to these threads being within a warp, no conditional statement about the thread ID's or any synchronization steps are required, as it is not possible for them to be out of sync. This results in a simple 6 step process (the last block of 'if' statements in the following example) that the whole warp computes in unison, achieving the summation of 64 elements.

```
//      Parallel sum a one-dimensional array arr[ ]
__syncthreads();
if (blockDim.x >= 1024)
{
        if (threadIdx.x < 512)
        {
                arr[threadIdx.x] += arr[threadIdx.x + 512];
        }
        __syncthreads();
}

if (blockDim.x >= 512)
{
        if (threadIdx.x < 256)
        {
                arr[threadIdx.x] += arr[threadIdx.x + 256];
        }
        __syncthreads();
}

if (blockDim.x >= 256)
{
        if (threadIdx.x < 128)
        {
                arr[threadIdx.x] += arr[threadIdx.x + 128];
        }
        __syncthreads();
}

if (blockDim.x >= 128)
{
        if (threadIdx.x < 64)
        {
                arr[threadIdx.x] += arr[threadIdx.x + 64];
        }
        __syncthreads();
}

if (threadIdx.x < 32)
{
        if (blockDim.x >= 64) arr[threadIdx.x] += arr[threadIdx.x + 32];
        if (blockDim.x >= 32) arr[threadIdx.x] += arr[threadIdx.x + 16];
        if (blockDim.x >= 16) arr[threadIdx.x] += arr[threadIdx.x + 8];
        if (blockDim.x >= 8)   arr[threadIdx.x] += arr[threadIdx.x + 4];
        if (blockDim.x >= 4)   arr[threadIdx.x] += arr[threadIdx.x + 2];
        if (blockDim.x >= 2)   arr[threadIdx.x] += arr[threadIdx.x + 1];
}
```

4.4.3 Interpolation

As hinted at earlier, the availability of texture memory on a CUDA device allows for rapid hardware accelerated linear interpolation. The flexibility of this interpolation process in dealing with edge conditions provides a crucial tool in optimizing processes which require interpolation. However, if more than a linear interpolation is desired additional work is required. Although cubic interpolation cannot be hardware accelerated it can still be performed very rapidly. By utilizing the special attributes of texture memory it is possible to retrieve all the required elements to perform a cubic interpolation with a single call to texture memory, due to the spatially local nature of a cubic interpolation. A minor extension to the cubic interpolation process can then be applied to allow for bicubic interpolation, ideal for interpolating values from maps.

4.5 CUDA mathematical processes

Many mathematical procedures which are frequently used in scientific computing have already been optimized in the C programming language and are well documented in Numerical Recipes (Press et al., 1992b,a). Using this source as a starting point, modified versions of these documented functions have been developed to work on a CUDA device.

4.5.1 Root solving

To determine the roots of a high order polynomial requires an iterative process. Numerous mathematicians have developed root solving procedures, each with different strengths and weaknesses depending on the nature of the problem. One such procedure is Laguerre's method. Although it may not be the fastest available root solving routine, it is robust and almost always guarantees convergence. The root solving method of Laguerre has been implemented in the C programming language by Press et al. (1992b), and is known as ZROOTS. We have developed a CUDA implementation of this code to remove C specific functions and adapt it to work with multiple threads.

To determine the image positions of a binary lens, a 5^{th} order polynomial needs to be solved. This is achieved using a code developed by Skowron & Gould (2012) which is based on the numerical root finding methods of Laguerre, Newton, Simpson (1740), and a new method described in their paper. Skowran and Gould present a unique implementation of the root solving methods specifically designed for use with inverting the binary lens equation. This method is quoted to be 1.6-3 times faster than the standard ZROOTS function presented by Press et al. (1992b). We have taken Skowran and

Gould's published Fortran code and converted it into C so that it can be implemented into CUDA for use on NVIDIA GPUs. We found that a CUDA implementation of Skowran and Gould's method was faster than a CUDA implementation of Numerical Recipes ZROOTS by up to a factor of 14.

In the root solving method developed by Skowron & Gould (2012) an additional feature has been specially developed for binary lens root solving. They provide an ability to use estimates of the root solution which allows the method to 'polish' them. This can produce rapid root solutions when you have a good initial estimate, which is especially useful for microlensing methods such as contour integration.

The 'polish' feature suggests it is possible to improve the performance in the production of high resolution magnification maps on a CUDA device, however, upon implementation it was found to produce slower results. This is possibly caused by a combination of two factors. Firstly, to use a magnification map's pixel root solution information, the solutions need to be stored and then transferred onto the CUDA device. This means 5 more large arrays have to be moved onto the CUDA memory. The time taken to move this large quantity of additional information is significant and has the result of negating possible time gains from only polishing the roots. Secondly, each thread in a warp is now polishing a solution. Depending on how far from the true solution the initial estimate is, it may cause multiple thread divergences as each thread in the warp has to polish the solution by a different amount. If no information is provided, every thread within a warp is more likely to follow the same path in determining its root solutions. Combining both these affects results in slowing down the root solving process to the point where any potential gains from only polishing the roots are no longer found. For these reasons, all threads solve the 5^{th} order polynomial each time without any prior information.

4.5.2 Simplex downhill

Determining the set of model parameters that best describe a set of data is a challenge that is frequently faced. A wide range of numerical methods exist that attempt to solve this problem, but to date, no solution exists that guarantees the determination of the true solution. One numerical method which uses a minimization technique, is the simplex downhill method (Nelder & Mead, 1965). Given an initial set of starting parameter sets (equal to $N_{dim} + 1$), the process explores the χ^2 surface and makes steps in the negative gradient of the parameter space direction, until it reaches a minimum. This method is simple to implement and successfully locates local minima in the parameter space, but it is not without limitations. As it only works towards the downhill gradient of the parameter space it is prone to getting stuck in local minima, and is frequently

unaware of possible global minima in the wider parameter space. Should the parameter space be very uneven, the simplex downhill approach struggles to determine the direction of the global minimum. These challenges are not unique to simplex downhill and techniques exist in all minimization methods to try and overcomes these hurdles. A commonly used computational minimization routine based on simplex downhill, but with an adaptive step size is called AMOEBA and has been developed in the C programming language by Press et al. (1992b). As with the ZROOTS function detailed above, a CUDA implementation of this C code has been developed to work on an NVIDIA GPU.

4.6 Summary

The use of GPUs in science is a new area of research, however, major advances have already been made in numerous other fields. To ensure this alternate computational tool is utilized to its best ability, an understanding of what it does and how it works is required. This chapter presents a comparison of the conventional CPU and a GPU to establish the main differences between these devices. An overview of a NVIDIA CUDA device's architecture is shown, detailing its advantages and use of its unique attributes. Provided with this alternate set of tools, optimized adaptations of common computational tasks are presented for use with a GPU, which achieve impressive performance gains compared with serial CPU methods. Additionally the optimization of specific proceduers useful in minimization methods are shown, including the CUDA implemented root solving procedure of Skowron & Gould (2012), which achieves a performance increase of up to 14 times.

Chapter 5

Modelling approach

5.1 Outline

This chapter details modelling stratergies, the methods developed for determining the best parameter solutions for single lens points source and binary lens models, and justifications of the alternate methods presented.

5.2 Modelling methods

To determine the model parameters that best describe a data set, a test to determine how well any given model fits the data is required. A process is then needed to try different model parameter sets and locate the best fitting solution.

5.2.1 Bayesian inference

An important method used in modelling that differs from the standard approach, is the use of Bayesian inference and a mixing model. A bayesian approach to modelling data varies from the commonly used frequentist method (Sivia, 2006), with the inclusion of prior information used in the probability of the model (D. Gamerman, 2006). An approach that can be used to rule out any physically impossible parameters, or model the likelihood of the parameters value, is Bayes' theorem. This can be derived from a standard rule of probability, the product rule (Sivia, 2006). If $P(A|B)$ means the probability of event A given event B has occurred, then

$$P(X,Y|I) = P(X|Y,I)P(Y|I) = P(Y|X,I)P(X|I) \ . \tag{5.1}$$

where I is the relevant background information. A simple re-arrangment now leads to Bayes' theorem,

$$P(X|Y,I) = \frac{P(Y|X,I)P(X|I)}{P(Y|I)} \quad . \tag{5.2}$$

Consider the case of observed data and a proposed hypothesis, from Bayes theorem,

$$P(hypothesis|data,I) \propto P(data|hypothesis,I) \times P(hypothesis|I) \quad . \tag{5.3}$$

Each of these terms have recognized names which will be used hereafter, $P(hypothesis|I)$ is the prior probability, a value which indicates the prior knowledge about the proposed distribution independent of the real data. $P(data|hypothesis,I)$ is known as the likelihood function. Together these values produce $P(hypothesis|data,I)$ the posterior probability, a measurement of the probability of the hypothesis given the data and the prior information. It is noted that $P(data|I)$ (sometimes known as the evidence) is omitted in Equation 5.3. This is acceptable in cases where parameter estimation is being performed as it is a normalization constant.

5.2.2 Mixing model

The posterior probability is directly proportional to the likelihood function

$$\mathscr{L} \propto e^{-\frac{\chi^2}{2}} \quad . \tag{5.4}$$

where $\chi^2 = \sum_{i=1}^{N} \frac{(\text{data}-\text{model})^2}{\sigma^2}$. Usually any outliers will have a significant effect on the model's posterior probability. However, instead of removing these data points, the model can be modified to include these bad data points but become less sensitive to them. Following the prescription of Hogg et al. (2010) it is possible to produce a mixing model that achieves this. It requires the addition of a binary integer per data point, which equals 1 if the data point is good, or 0 if it is a bad data point. As well as the inclusion of three more parameters, P_b the prior probability of any data point being bad, Y_b the mean of the bad data distribution and V_b its variance. This large number of parameters is not computationally feasible, therefore through marginalization the model is reduced from having more parameters than data points, to an approximate likelihood function with $j+3$ parameters, (where j is the number of original model parameters),

$$\mathscr{L} \propto \prod_{i=1}^{N} \left[\frac{1-P_b}{\sqrt{2\pi\sigma_{y_i}^2}} \exp\left(-\frac{[y_i - f(x_i)]^2}{2\sigma_{y_i}^2} \right) + \frac{P_b}{\sqrt{2\pi\left[V_b + \sigma_{y_i}^2\right]}} \exp\left(-\frac{[y_i - Y_b]^2}{2\left[V_b + \sigma_{y_i}^2\right]} \right) \right] . \tag{5.5}$$

where y_i is the observed data at a point x_i, and a model which depends on x, $f(x)$.

5.2.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a procedure which trials numerous parameter sets testing each likelihood, which it then uses to determine the next set of trial parameters W.R. Gilks (1997). This technique is commonly used throughout the world, from the simple game of battleships to complex computational calculations.

5.2.3.1 Metropolis algorithm

The Metropolis algorithm controls how the parameter space is sampled (Metropolis et al., 1953; D. Gamerman, 2006). It requires a set of initial guess values for each Markov chain parameter, as well as an initial step size (σ) value for each. These inputs enable a random walk search of all parameter values simultaneously.

Our implementation of the Metropolis Hastings MCMC uses multiple chains that are distributed around the initial guess parameters, $U_{j,1}$, where j defines each parameter. The starting point of each chain is calculated as, $U_{j,1} + \alpha\sigma_j$, where α is a random value between -1 and 1, which is repeated for each parameter (j), for each chain. The likelihood of the first point in each chain is calculated as $\mathscr{L}_1$ using Equation 5.5, which is stored in an array of $\mathscr{L}_i$. Each chain makes its next step by taking its previous point ($U_{j,i-1}$) and adding $\alpha\sigma_j$ for each parameter. The new set of parameters ($U_{j,i}$) is used to calculate a new likelihood. A test is performed to determine if the newly proposed set of parameters is more likely ($\mathscr{L}_i > \mathscr{L}_{i-1}$). If true, the Markov chain stores this parameter set as the new point and repeats the process of selecting the next new parameter set from this point. Should the previous likelihood be greater than the proposed parameter set ($\mathscr{L}_i < \mathscr{L}_{i-1}$), a second test is used to decide if this less likely solution should still be kept or discarded. Taking a random value between 0 and 1, γ, a test of $\gamma \leq \frac{\mathscr{L}_i}{\mathscr{L}_{i-1}}$ is used, if true this less likely parameter set is kept and stored as the next new point in the Markov chain. Alternatively, if $\gamma > \frac{\mathscr{L}_i}{\mathscr{L}_{i-1}}$ the proposed parameter set is discarded and

the previous set of parameters which produced $\mathscr{L}_{i-1}$ is kept and stored as the new point in the chain. This process is repeated for as long as desired, producing a chain of results from which a new step size can be calculated, or a set of most probable parameters can be derived from the sampled parameter space.

5.2.3.2 Adaptive step size

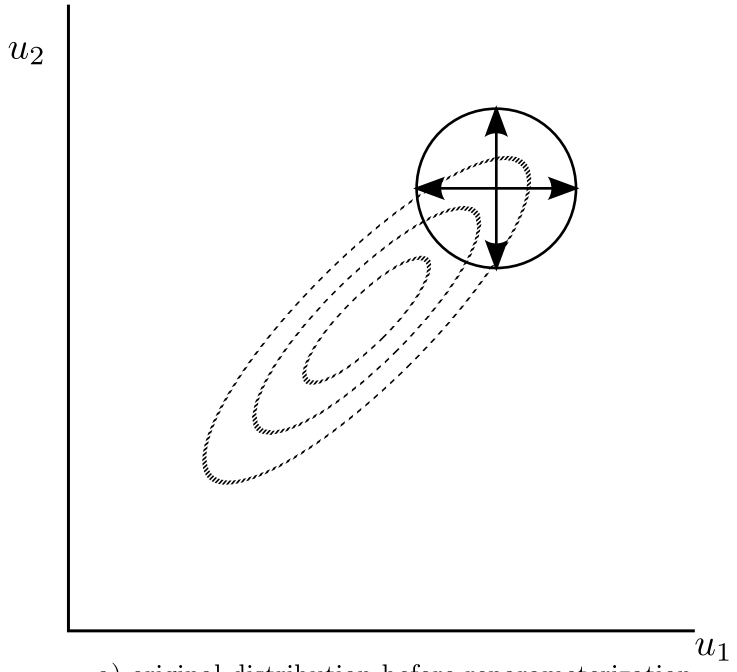
An appropriate step size is essential for the convergence of multiple chains in a suitable time. If the step size is too large the sampling will frequently reject parameters that are too far from the solution, however, if the step size is too small it will be slow to migrate as it has to take many steps before it reaches areas of high likelihood. An important reason for using an adaptive step size is to improve the convergence of multiple chains in a non-uniformly distributed parameter space, which results in a minima being located. This is achieved by reparameterising the distribution it is sampled from so it is along its principle axis, which comes from its covariance matrix.

Figure 5.1 shows that for situation a) the sample area (solid circle) lies mostly outside the main diagonal of likely solutions (dashed contour lines), where as in situation b), after re-parameterization, the sample covers a greater area of parameter space which has a greater likelihood. Due to this increased overlapping area, a randomly chosen parameter set has an increased chance of having a larger likelihood. Following the prescription of Doran & Müller (2004) outlined below, an adaptive step size Gaussian sampler is developed to select new step sizes for each parameter and transform the parameter space for each chain's steps.

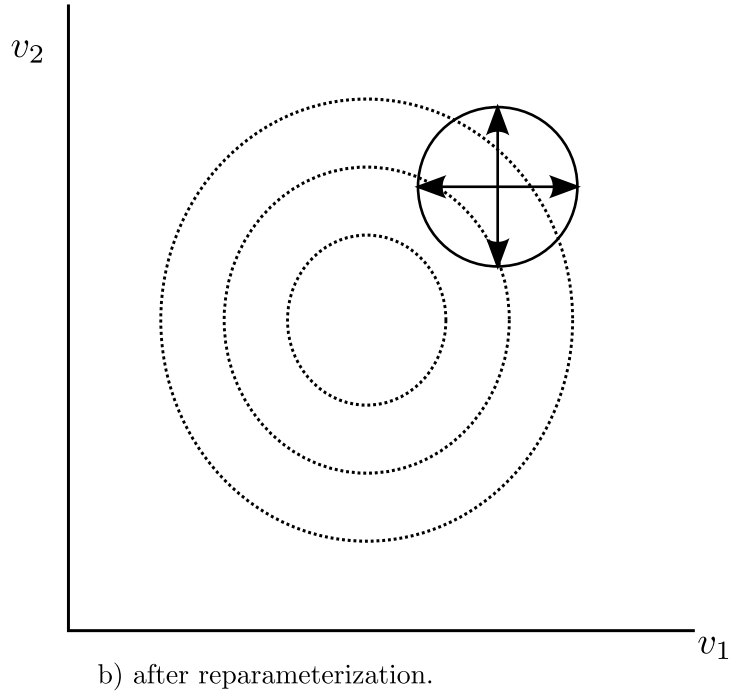
To calculate the new sample using an adaptive step size, a matrix of accepted chain parameters (chain iterations, i, by parameters, j) is used. The covariance matrix is computed and its eigenvalues λ_j^2 are determined, which become the new step size. The eigenvectors are also found and used as a transformation matrix, T. To make a new parameter selection, the original parameters (u_i) are transformed by T ($v_i \equiv T^T u_i$), from which a new Gaussian sample (v_{i+1}) can be taken using the new step size, $v_{i+1,j} = v_{i,j}\lambda_j$. This produces a new sample with the transformed matrix and is converted back to the original parameterization using the transformation matrix again ($u_{i+1} = Tv_{i+1}$).

5.2.3.3 Convergence test

MCMC steps are repeated until all the chains are considered to have converged to a common area of parameter space. To determine when this has occurred the following test developed by Gelman & Rubin (1992)(Doran & Müller, 2004) is used:



a) original distribution before reparameterization.



b) after reparameterization.

FIGURE 5.1: A graphical representation of the MCMC re-parameterization performed, enabling an adaptive step size sampler to help convergence to the global minimum in fewer steps for non-uniformly distributed parameter spaces. The dashed line shows the contours of likelihood and the solid circle is the sample area of an MCMC step.

$$R_j = \frac{n-1}{n} + \frac{B_j}{nW_j} \quad , \tag{5.6}$$

where R_j is the convergence test value for each parameter, j, which tends to 1 as the chains converge, $2n$ is the length of the chains, B_j is the variance in the mean between chains for each parameter and W_j is the variance within the chain for each parameter. For a multiple parameter test, every R_j value has to be less then the acceptance threshold (set as $R_j < 1.1$) for the chains to be considered converged. To accurately determine the final solution a burn in is initially performed, before a set of sample chains around the global minimum are initialized. The burn in is a process where the chains explore a wider parameter space, locating areas of local minima before they should all converge to the global minima. This is an important stage to help confirm that the minima located is the most likely. Typically this is the slower stage as it explores a larger parameter space. Once complete it is possible to initialize a new set of chains with a small distribution around the area of global minimum with confidence that they are not stuck in local minima. Performing a second sample with a smaller distribution located around the minima allows for finer movements in the parameter space. It attempts to accurately determine the minimum to a greater accuracy and put estimations for their variances.

5.2.4 EMCEE — A python Markov Chain Monte Carlo module

Given an appropriate set of starting values for the Markov chain, the MCMC method successfully converges to a suitable model for single lens events, however, it can take numerous steps and is not always rapid in its convergence. Although it uses the adaptive step size approach detailed in section 5.2.3.2 to reduce the number of steps required and therefore the time taken. The increased processing that these transformations require increases the length of time each step takes. Numerous Markov chain optimization approaches have been developed, many that are based on the Metropolis-Hastings algorithm. One such method is the t-walk developed by Christen & Fox (2007). It uses two independent points and performs a generic, self adjusting MCMC. This method later inspired the work of Goodman & Weare (2010) who developed an MCMC method that is rapid and performs well in highly correlated parameter space. It utilises ensemble samplers consisting of numerous "walkers" which step through parameter space one at a time based on partial resampling (Liu, 2008) and Metropolis-Hastings rules. The walkers each move using a "stretch move", defined in (Goodman & Weare, 2010), which is affine invariant. This MCMC method is efficient and works well in all situations, including highly anisotropic distributions. Due to the efficiency of this method it has been developed as an all inclusive python module, EMCEE, by Foreman-Mackey et al. (2013). It

requires minimal input from the user, enabling it to perform a thorough MCMC search of parameter space. The code is easy to implement and its success is apparent, being used in published astrophysical literature (Watkins et al., 2013; Boettcher et al., 2013; Zhou & Huang, 2013; Fraser et al., 2013). The only requirements to start an EMCEE search are an initial distribution of walkers, the number of dimensions and desired steps in the chains, and to provide a likelihood function.

5.3 Single lens modelling

The majority of microlensing events are caused by single lenses, meaning, before they are modeled as a binary lens they should be investigated as a single lens. Current publicly listed single lens models are not generated using preferential methods, therefore, I present here an alternate approach to modelling single lens events.

5.3.1 How this approach is different

To understand how this approach is different from what is already being used and why there is a desire to change an established modelling process, the operation of current modelling methods need to be understood. Going back to the initial data acquisition from the observatories, sets of flux or magnitude measurements over time are provided for each microlensing event, these can be requested for use for modelling. Upon retrieval of the data, a common practice is to perform a data rejection processes to discard any 'bad points'. This is the crucial concept that our alternate approach avoids. Data rejection processes are not without fault as they run the risk of discarding potentially good data points. Furthermore, potential problems arise from the data rejection methods used. Methods such as sigma clipping and kappa clipping introduce undesirable biases. Both these methods assume an initial model for the data, from which the bad data points are determined and removed. Producing a new set of data to work with, to then improve the model further. This technique has the affect of self-reinforcing the initially assumed model. Alternatively, data can be chosen manually as bad and removed by the user, this is the least desirable method as it is subjective and therefore irreproducible. As shown in these examples, removing data for the modelling process is not without fault and an alternate approach is beneficial.

5.3.2 Modelling process

The following method outlined does not use any data rejection techniques; instead it models a single lens event using the entire set of data, good and bad. To achieve this it utilizes MCMC (W.R. Gilks, 1997), Bayesian inference (Sivia, 2006), as well as a mixing model (Hogg et al., 2010).

5.3.2.1 Single lens model

Using the single lens model defined in section 2.3.1, a set of three parameters describe the shape of the curve. To test how well each model fits the data a mixing model is used,

$$\mathscr{L} \propto \prod_{i=1}^{N} \left[\frac{1-P_b}{\sqrt{2\pi\sigma_{y_i}^2}} \exp\left(-\frac{[\Delta F_i - F_0 A(u_0, t_0, t_E, t_i) + F_R]^2}{2\sigma_{y_i}^2} \right) + \frac{P_b}{\sqrt{2\pi\left[V_b + \sigma_{y_i}^2\right]}} \exp\left(-\frac{[\Delta F_i - Y_b]^2}{2\left[V_b + \sigma_{y_i}^2\right]} \right) \right] . \tag{5.7}$$

5.3.2.2 Priors

Equation 5.3 shows that a probability for the parameters can be determined from prior information and a likelihood function. In our case, the priors of each model parameters are set to:

$$\begin{aligned} P(u_0|I) &= \begin{cases} 1 & 1.6 \times 10^{-}6 < u_0 < 8.0 \\ 0 & \text{otherwise} \end{cases} \\ P(t_0|I) &= \begin{cases} 1 & t_0 > 0 \\ 0 & \text{otherwise} \end{cases} \\ P(t_E|I) &= \begin{cases} e^{-\left(\frac{\log(t_E)-1.342}{0.342}\right)^2} & t_E > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.8}$$

where the prior distribution for t_E comes from the average expected Einstein time based on the distribution of lenses within our galaxy (Sumi et al., 2011).

To optimize the computation of single lens events, the mean and variance of bad data are defined to be constant, therefore there are two less parameters to model. In an ideal case there would be no scatter in the data (i.e. $Y_b = 0$). Due to the large number of baseline data scattered either side of zero, any noise should average to zero. For this reason the mean error value is taken to be zero ($Y_b = 0$). In reality there is noise in all events, even those of very large magnification, therefore a large variance is required. For a point source lens the magnification can be infinite, so a large variance of $V_b = 10^9$ is chosen. By forcing these two parameters to be constant, a significant reduction in computational time is achieved, as there is only one additional parameter to model, P_b. This is included in the bayesian modelling with a prior of,

$$P(u_0|I) = \begin{cases} 1 & 0 \leq P_b \leq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

5.3.2.3 Initial guess parameters

The initial guess parameter values need to be carefully chosen to aid in the convergence of Markov chains. Appropriate starting points need to be chosen so that the initial models are not too far from the true solution. This would make it difficult and time consuming for the chain to move large distances towards a global minimum. Underestimating the width of a single lens peak is preferential to overestimating, so that the MCMC step selections attempt to fit light curves with narrow peaks close to the data before gradually spreading out into a wider sample space. This has the effect of the peak being allowed to grow into the shape of the data, rather than start larger and risk the potential of it growing outwards and away from the data, preventing it from ever converging. u_0 and P_b are both parameters that can only have values between 0 and 1. As this is a small parameter space, the initial selection of all u_0 values are chosen as a random value between 0 and 1, with a step size in the Markov chain that would allow it to freely move around this range. P_b represents the probability of bad data, it is assumed that the data is mostly good and therefore an initial guess parameter is chosen to be a random value between 0-3% of the data being bad. This does not constrain the parameter to remain in this range and should the data be worse, the parameter quickly increases to reflect this.

5.4 Single lens parameter space exploration

The procedure used to search parameter space of single lens events was based upon the Metropolis Hastings algorithm and adaptive step size detailed in section 5.2.3. However, the implementation of the documented python EMCEE tool which uses an affine invariant ensemble samples was found to be significantly faster and converged more regularly in situations where previous modelling methods struggled. This resulted in the EMCEE tool being the parameter search method used.

A thorough search of parameter space is performed using EMCEE by completing a burn-in process to locate the minimum, followed by a sample to explore the global minimum space, from which a well converged solution is determined. Figure 5.2 shows a typical sample distribution for a post burn in trial. Well defined global minima of each parameter are apparent, where samples are usually distributed in a gaussian fashion. On some occasions a pair of comparable local minima are found (Figure 5.3). Therefore, instead of taking the mean of the distribution as the model solution (solid red line), the parameter set that produced the maximum likelihood (solid green line) value is recorded. In most cases these parameter sets are the same, but in situations where there are multiple peaks in histograms, the most likely solution is at one of the peaks, unlike the mean value which may fall into an area of low likelihood.

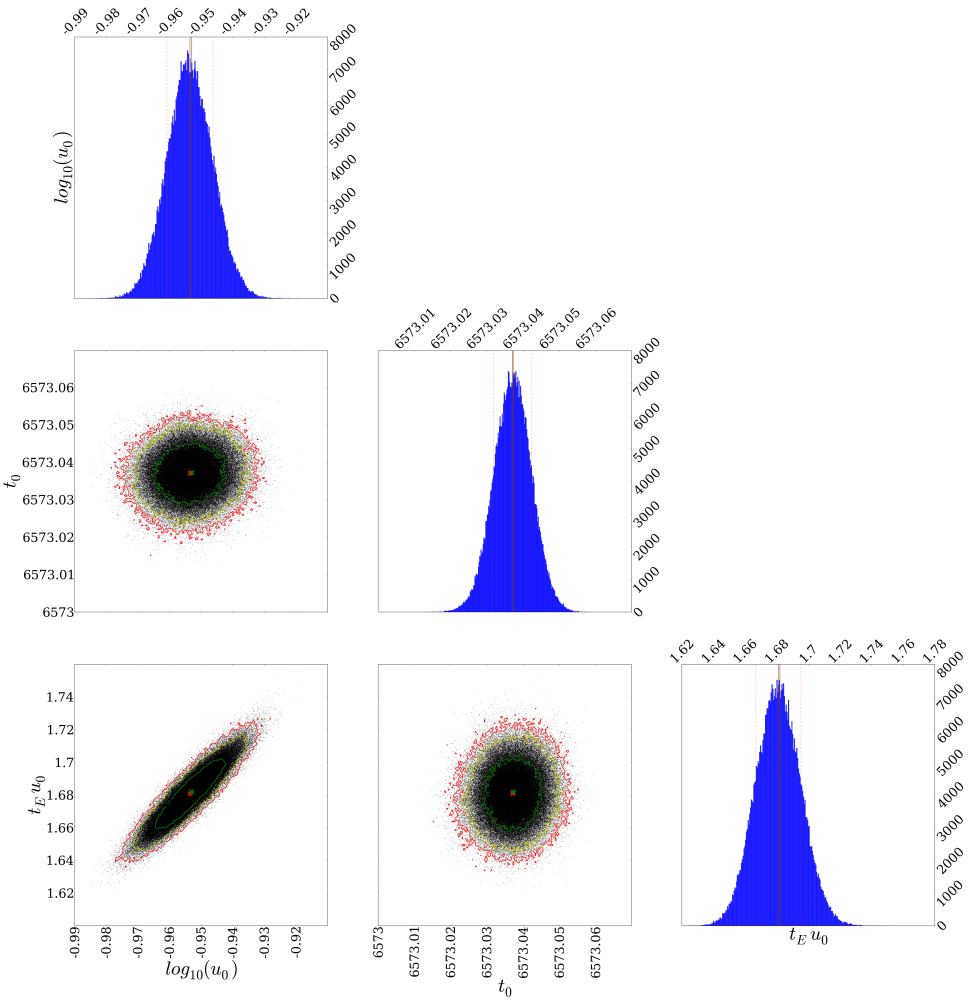


FIGURE 5.2: The output from a completed EMCEE process clearly showing a single gaussian peak, with the fitted mean (red lines and crosses), standard deviation (dashed red lines) and highest likelihood recorded (green lines and crosses) solutions.

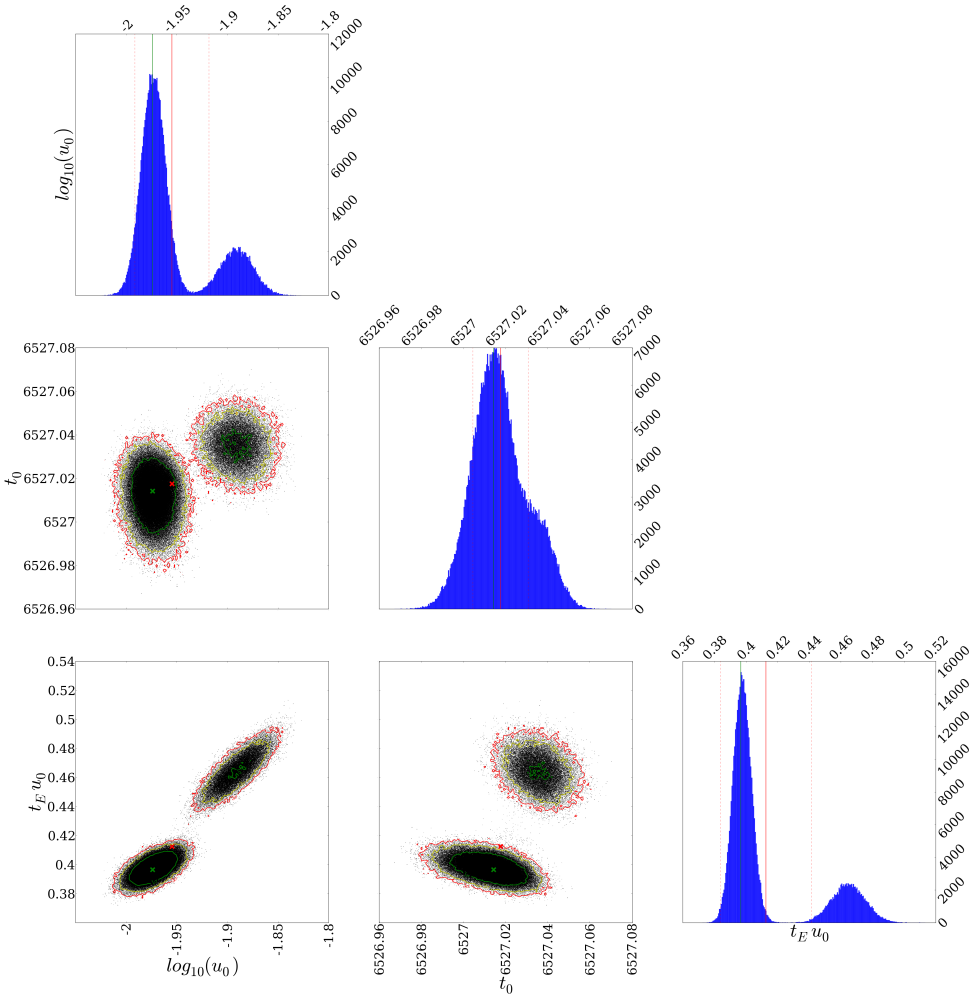


FIGURE 5.3: The output from a completed EMCEE process showing a pair of peaks, with the fitted mean (red lines and crosses), standard deviation (dashed red lines) and highest likelihood recorded (green lines and crosses) solutions. The movement of the mean away from one of the peaks is clearly shown.

5.5 Binary lens modelling

Using a combination of python and CUDA programming languages via a python module PyCUDA (Klöckner et al., 2009), allows relatively easy access to the functionality of GPU cards manufactured by NVIDIA. The CUDA language is primarily written in C/C++, but contains some subtle differences.

A grid and MCMC parameter search routine has been developed to work on CUDA devices. It can take any binary lens event's data set and rapidly determine the most likely model parameters that fit the data.

5.5.1 A new modelling approach

In utilizing the performance gains of an NVIDIA Tesla C2075 GPU, new binary lens modelling techniques can be used. These methods have been adapted to work on a CUDA device and are unfeasible to implement on a CPU. Our new modelling method involves a mix of commonly used techniques for binary lens modelling, the direct analytical approach, map interpolation and inverse ray shooting.

Directly solving the binary lens equation by using a GPU enables an accurate point source point lens magnification map to be produced, an approach that would have been unfeasible without the parallel possibilities of a GPU. With an accurate magnification map, trajectories can be interpolated across it, producing light curves. Interpolated trajectories show advantages in speed when comparing multiple trajectories, as minimal additional calculations are required. Producing the magnification maps by directly solving the binary lens equation has the advantage of high precision maps in a few steps. Additionally, the use of magnification maps enables any convolution process, allowing for the exploration of finite sources (ρ) with minimal additional work.

With a convolved magnification map generated from the three parameters d, q, ρ, it is possible to interpolate a trajectory across it, which is determined by u_0 and ϕ. Producing the distinct shape of a binary lens light curve, which can then be scaled in time by t_0 and t_E. This approach determines a light curve of any given binary lens parameter set.

5.5.1.1 Point source magnification map

On a standard CPU producing a directly solved magnification map is unfeasible, as the 5^{th} order polynomial solving routine is slow, and performing it in series millions of times takes too long. This problem can be overcome if developed in a parallel way, as

can be achieved with GPUs. Using multiple threads per block, we assign a single pixel of the magnification map to individual threads of each block. There are not enough threads available in a block to produce the whole magnification map in a single step, but it is possible to have a very large number of blocks. Meaning the pixels need to be divided up into areas which can be assigned to separate blocks. Optimum performance was achieved via experimentation and experience. Utilizing the maximum number of threads per block (1024 on a Tesla C2075) does not guarantee the fastest result. In general we have found that running fewer threads and more blocks can achieve faster results, most likely due to memory throughput limitations.

We have written a CUDA code that produces a magnification map of 1000×1000 pixels. A single block uses 500 of the available 1024 threads to calculate half a row of pixels of a magnification map, and the grid is made up of 2×500 blocks. The code reads in the magnification map dimensions, and the mass ratio and separation parameters (d, q). Each thread computes its appropriate pixel position in the lens plane, ζ, and calculates its own point source image positions (z), by solving the binary lens equation,

$$\zeta = z - \frac{\varepsilon_1}{\bar{z} - \bar{z}_{m_1}} - \frac{\varepsilon_2}{\bar{z} - \bar{z}_{m_2}} \quad . \tag{5.10}$$

The optimized root solving method (section 4.5.1) is used to obtain each thread's five unique solutions. These are tested to see if they are true solutions by substituting them into the original equation (Equation 5.10). The solutions are accepted as true if they match the source position within a tolerance of 1×10^{-4}. This test criteria needs to be carefully considered, as a tolerance too small can prevent correct solutions being accepted and too large would accept invalid solutions, both of which cause erroneous values in the magnification map. With the correctly identified (3 or 5) roots, the magnification of the images are calculated (section 2.4.1) and added together to determine the total magnification of each source pixel.

From an understanding of binary lenses, it is known that all binary lens magnification maps are symmetrical about at least one axis, meaning only half of a map needs to be calculated. The code has been designed so that a two dimensional grid of blocks on the CUDA device are used, where one dimension is the total width of the magnification map divided by the number of threads per block, and the other dimension is half the total height of the map. Due to the symmetry of the maps, only half the height is calculated and is mirrored onto the other side to produce a completed map. It is found that instead of copying the whole array after its calculation, it is faster for each thread to store the pixel's value in its appropriate symmetrical array location directly after computation. The map is built by all threads calculating individual pixels and storing them in a

two dimensional global memory array, which is outputted from the GPU device upon completion. A flow chart showing the logic of this process can be seen in Figure G.1.

This routine has been found to be exceptionally fast and is able to produce a high-density magnification map in a fraction of the time it takes on a CPU. Producing the same resolution (1000×1000 pixels) double precision magnification map on an Intel i7 CPU takes a little under 2 minutes, whereas on a Tesla C2075 GPU it is over 1200 times faster, taking only 0.09 seconds. This shows that a GPU routine improves performance and generates fast, high-resolutions magnification maps, as seen in Figure 5.4.

5.5.1.2 Higher resolution magnification maps

To include finite source effects, the magnification map needs to be convolved. This requires the magnification map to have a pixel size smaller than the source size, and for improved accuracy, the smallest pixel sizes are preferential. It is possible to achieve this by creating a high-resolution magnification map initially, however, as will become clear later, for large sources the convolution process would slow down significantly. Therefore an alternate approach is developed to achieve fast results dependent on the source size. For small sources, a magnification map of 1000×1000 is not of a high enough resolution to achieve a minimum of 10 pixels per source diameter. Instead of recomputing a full magnification map at the desired resolution, it is possible to use the information already computed in the 1000×1000 magnification map.

The first stage in the production of higher-resolution magnification maps is determining the required resolution. Taking the ceiling of the minimum desired resolution divided by the current pixels per source diameter, provides the number of sub-pixels that the current map needs to be divided up into, as shown in equation 5.11,

$$\left\lceil \frac{\text{minimum resolution} \times \text{map size/current map resolution}}{\text{source diameter}} \right\rceil . \tag{5.11}$$

With an appropriate resolution determined and the original magnification map loaded into texture memory, the thread and block dimensions of the CUDA device can be set up in the same fashion as with the original (1000×1000) magnification map. Each pixel in the higher-resolution map can be considered as a square group of sub-pixels of the original map. Determining the magnification at each sub-pixel follows the same process as before (section 5.5.1.1), but to save time some approximations can be made. As most pixels will be far from the caustic structures they will have a small magnification and magnification gradient. So instead of solving the 5^{th} order polynomial, it is acceptable

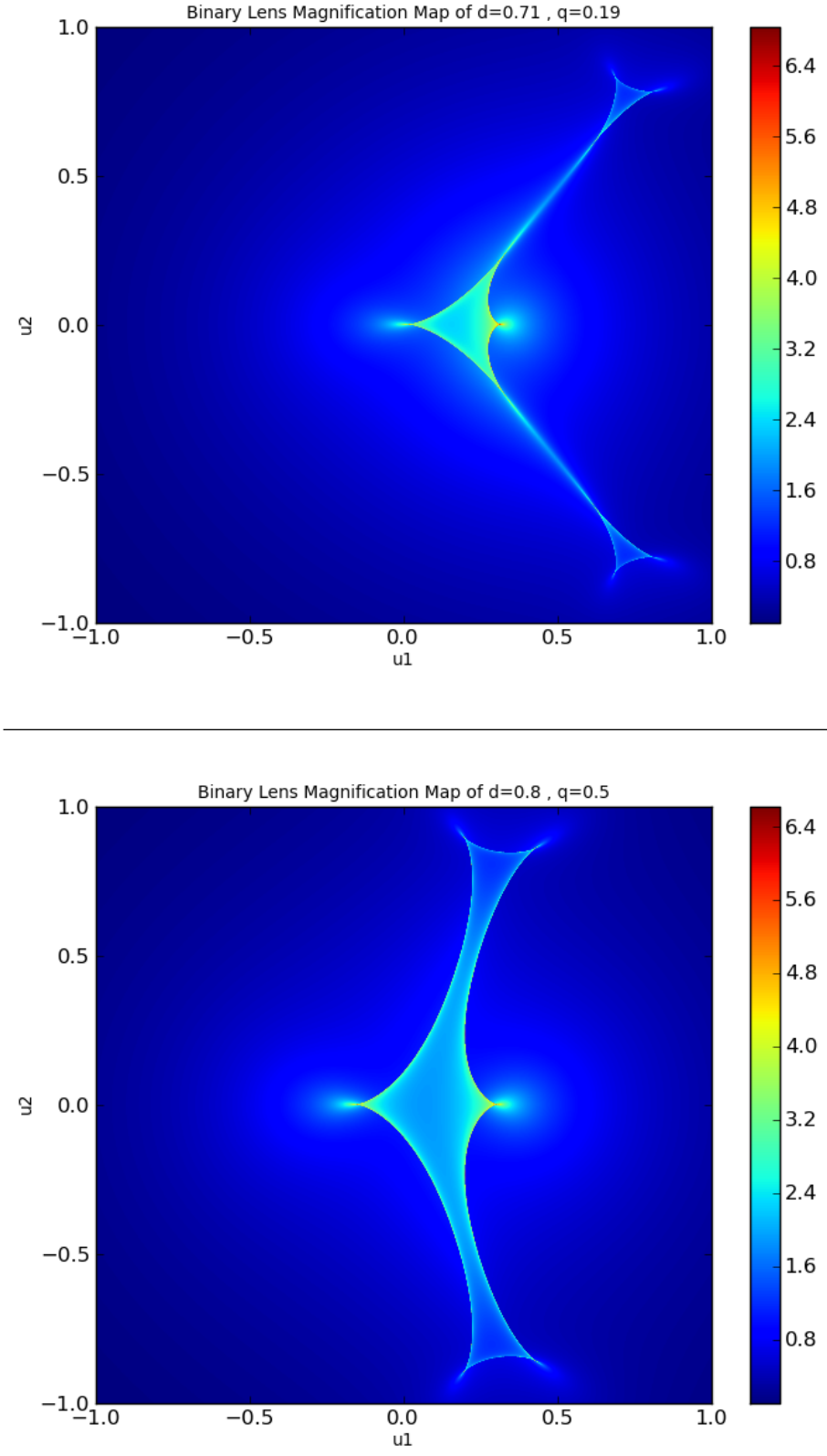


FIGURE 5.4: A pair of example magnification maps generated using the CUDA routine.

to interpolate the magnification value from the previously computed map. This is significantly faster, especially when utilizing the hardware accelerated bilinear interpolation. Should the interpolated magnification returned be deemed too high (> 5), then the 5^{th} order polynomial solution can be found following the same procedure used to create the original magnification map.

To avoid thread divergence as much as possible the threads are distributed on the CUDA device so that they are grouped together in areas on the magnification map. Large areas of pixels on the magnification map will require the same treatments and therefore each thread will follow the same path in the code (minimizing thread divergence). Rather than a scattered set of pixels where each thread will be required to perform different treatments. As for the initial production of the map it is still symmetrical, therefore only half needs to be computed and then mirrored across the symmetrical axis to reduce excess computational work. The steps followed to produce a higher-resolutions magnification map can be seen as a flow chart in Figure G.2.

5.5.1.3 High magnification pixel filter

When producing magnification maps the pixels are solved as if they are point sources, which means that on the caustic there is an infinite magnification. The gradient of magnification close to caustic edges is very sharp, therefore should the centre of a pixel on the magnification map fall very close to a caustic, it will return a very large magnification. However, as a pixel represents the magnification of the area it covers, these individually large magnification pixels are erroneous and cause problems when interpolating trajectories, as shown in Figure 5.5. Left unresolved, spots of high magnification can appear close to caustic edges, instead of an expected smooth transition. To prevent this from occurring, a high magnification filter is applied to the magnification map prior to convolution. This process would ideally use a median filter, however to compute the median is not computationally fast, so a mean filter is applied instead. The threads and blocks of the CUDA device are distributed in exactly the same way as they are for the generation of the higher-resolution magnification map and similarly the higher-resolution map being filtered is transferred to the texture memory of the CUDA device. Each thread is assigned an individual pixel of the magnification map, and by using spatially local texture memory requests, it rapidly retrieves the magnification values of the surrounding eight pixels from which an average is taken. Should the central pixel be greater than five times the average of its surrounding pixels, it is replaced by the average.

As with the higher-resolution magnification map generation, the distribution of threads avoids many thread divergent cases and computing only half the map then mirroring the

results, reduces computation time. Similarly, a flow chart showing the logical procession of applying this filter is shown in Figure G.3.

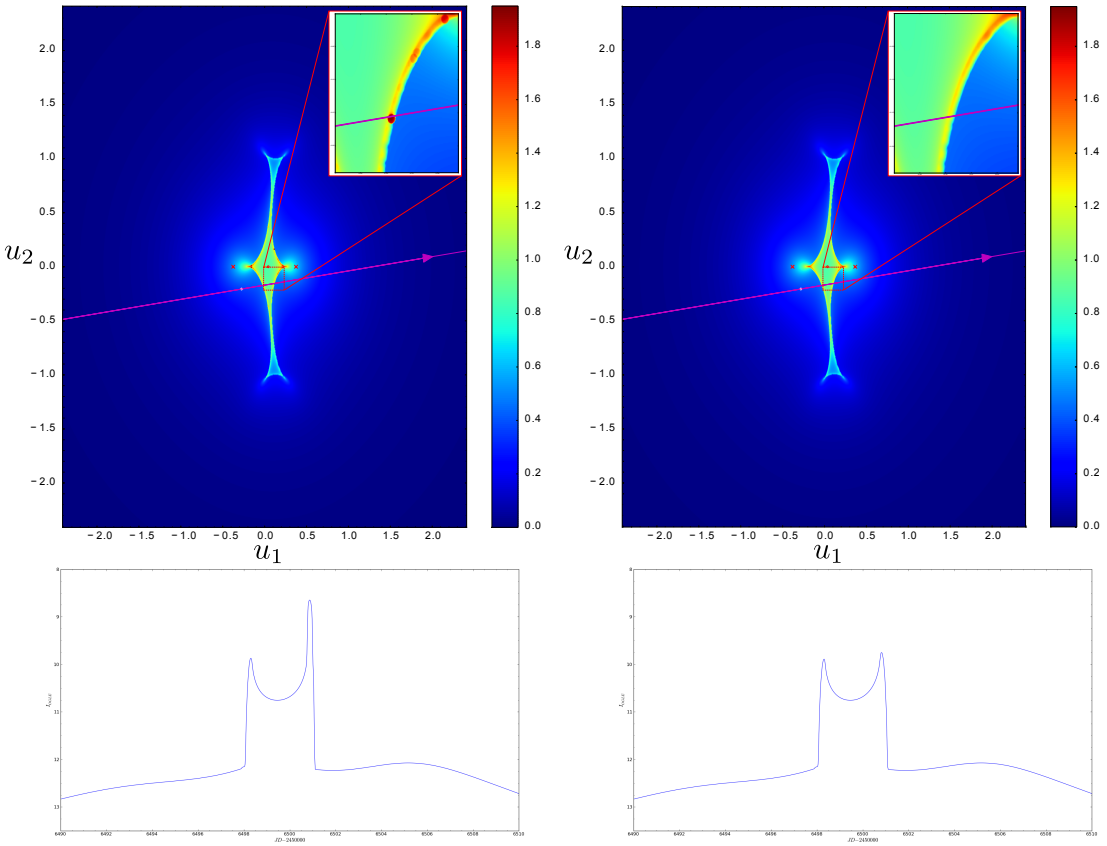


FIGURE 5.5: A filter method to remove individual very high magnification pixels.

5.5.1.4 Kernel calculation

A kernel is a square matrix equal to or just larger than the diameter of the finite source. It can be determined by aligning its centre with the source, and for each of its pixels calculate the fraction of the source enclosed. Determining the fraction enclosed when the source boundary passes through the pixel requires several conditional statements and integration.

Computing multiple decisions and solving an equation is not computationally fast, however, a faster but less accurate approach is possible. The multi threaded capabilities of the GPU enables a brute force approach to be used. For a single pixel of the kernel, a large number of threads on a GPU can be used to perform many simple tasks, instead of fewer complex processes. These threads can be assigned to a square array of sub-pixels within the kernel pixel (GPU block). Each thread then performs a simple comparison of its sub-pixel's distance from the centre of the kernel with the radius of the source,

assigning a value of 1 or 0 for inside or outside respectively. Figure 5.6 shows a kernel pixel encompassing part of a source (green line) where the blue circles are inside the source and return a 1, the red crosses are outside of the source and return a 0. By using multiple sub-pixels (threads), the source's percent enclosed by each kernel pixel (block) can be determined from the fraction of sub-pixels (threads) that lie within the source boundary. Despite the additional sums, the GPU's ability to perform all these simple calculations in parallel results in the entire kernel generation routine taking less than 0.005s (an example of which can be seen in Figure 5.7). Due to the large number of threads and how fast they can perform these calculations, of the 0.005 seconds taken most of this time is spent transferring the input and output memory between the GPU and host. For the kernel to be of use in convolving magnification maps it needs to be normalised, which is achieved by dividing each pixel by the sum of all pixels in the kernel.

Convolution kernels are circular as shown in Figure 5.7, therefore two axis of symmetry exist. Only computing a quarter of the kernel is therefore required, as the rest can be mirrored, shown in Figure 5.8.

5.5.1.5 Limb-darkening

To include limb-darkening into the kernel generation, a minor adaptation of changing each sub-pixel from returning a value of 1 for inside the source, with the result of a linear limb-darkening function (Equation 2.12), can be used. In Figure 5.6 the red crosses are outside of the source (green line) and therefore return 0, but the blue circles return a value I based on the sub-pixels distance from the centre of the source. These sub-pixels are again summed up and normalised, providing a limb-darkened percent enclosed of the source.

5.5.1.6 Map convolution

The convolution process multiplies each of the normalized kernel's pixel values by the corresponding magnification map's pixels, these are then summed to determine a total magnification.

Threads and blocks of the CUDA device are distributed in the same way as they were to produce the high-resolution magnification map, with the high-resolution map being transferred to the CUDA device texture memory. Each thread represents a single pixel on the convolved map (of the same dimensions as the high-resolution magnification map).

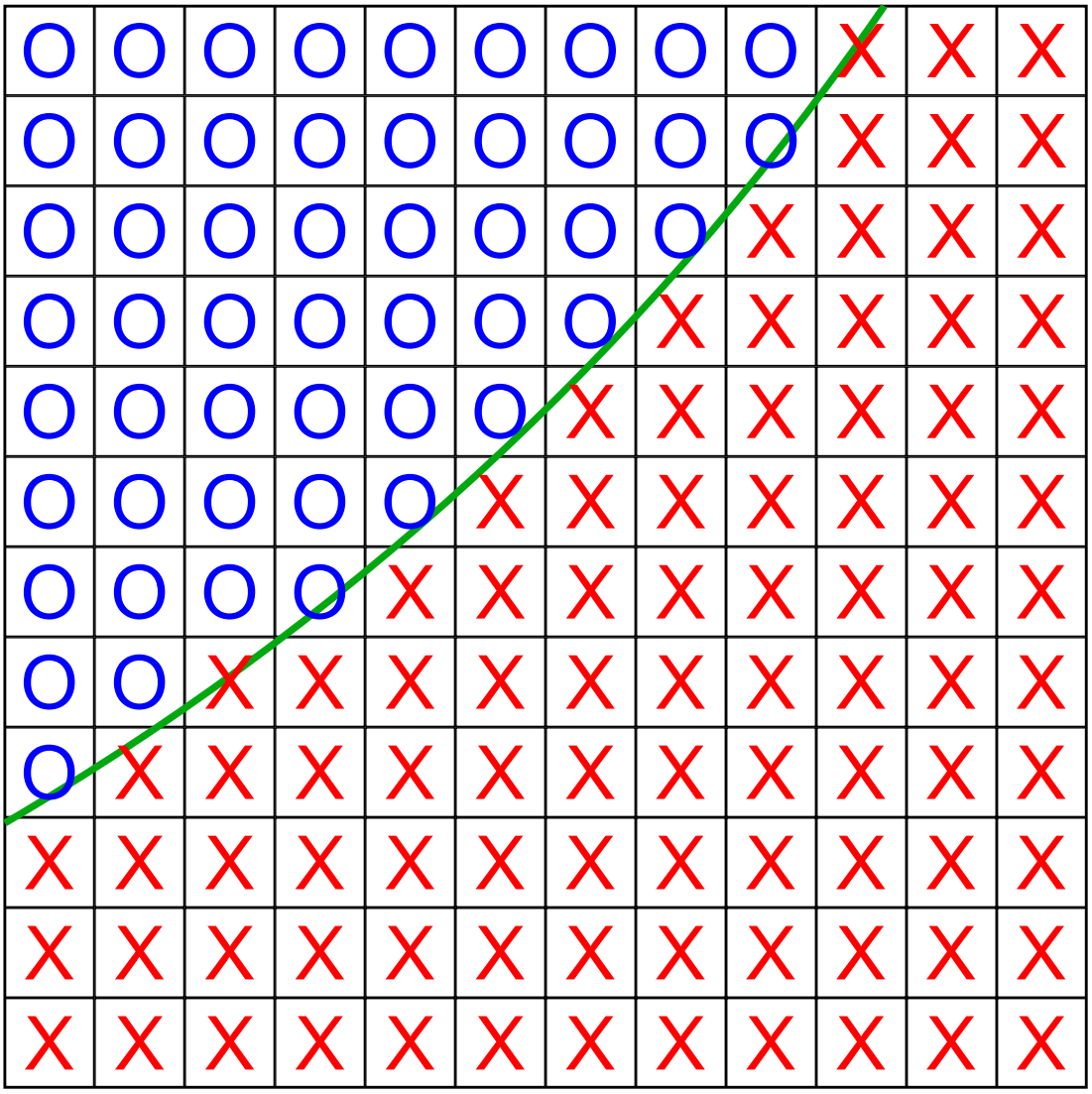


FIGURE 5.6: A single pixel of the kernel which is enclosing the edge of the source (green line). The blue circles indicate the sub-pixel lies inside the source, and the red crosses indicate the sub-pixel falls outside of the source. The total number of blue circles divided by the total number of sub-pixels provides the percentage of the source enclosed.

Every pixel is treated in the same way on separate threads which leads to the production of a full convolution map, showing the magnification achieved from a finite source at each pixel.

This convolution process is ideally suited for the use of texture memory, as each thread is always making spatially local calls to the texture memory. This is an optimum situation and allows full utilization of the CUDA architecture for fast memory accesses. Due to the pixel dimensions of the kernel being dependent on the resolution of the magnification map and the finite source size, it is not possible to dynamically retrieve every required pixel for a convolution calculation from the magnification map in a single step, instead

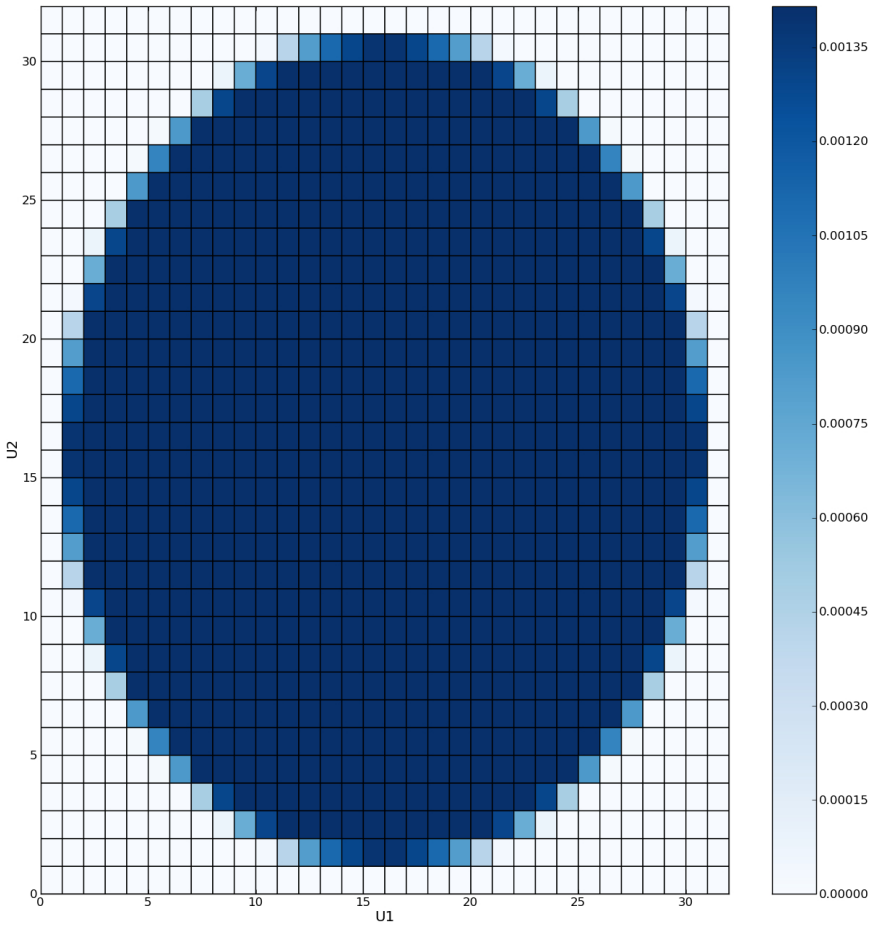


FIGURE 5.7: A 32x32 pixel kernel used in convolving a magnification map.

loops need to be used. This reduces the ability to perform spatially local memory accesses.

Techniques such a loop unrolling have been used to optimize the convolution process of known kernel dimensions, which is particularly helpful for small sources. When dealing with very large sources, if the initial magnification map resolution remains the same then the size of the kernel becomes very large. The production of a large kernel is not very time consuming, but its use in convolving the magnification map is. Typically the kernel has dimensions of 10×10 pixels, but it can grow to larger than 100×100. When the map is convolved each pixel would have to sum $100 \times 100 = 10,000$ values, compared with $10 \times 10 = 100$ that is typically required. Every pixel in the magnification map requires convolving and the map has a minimum resolution of 1000×1000, so the large

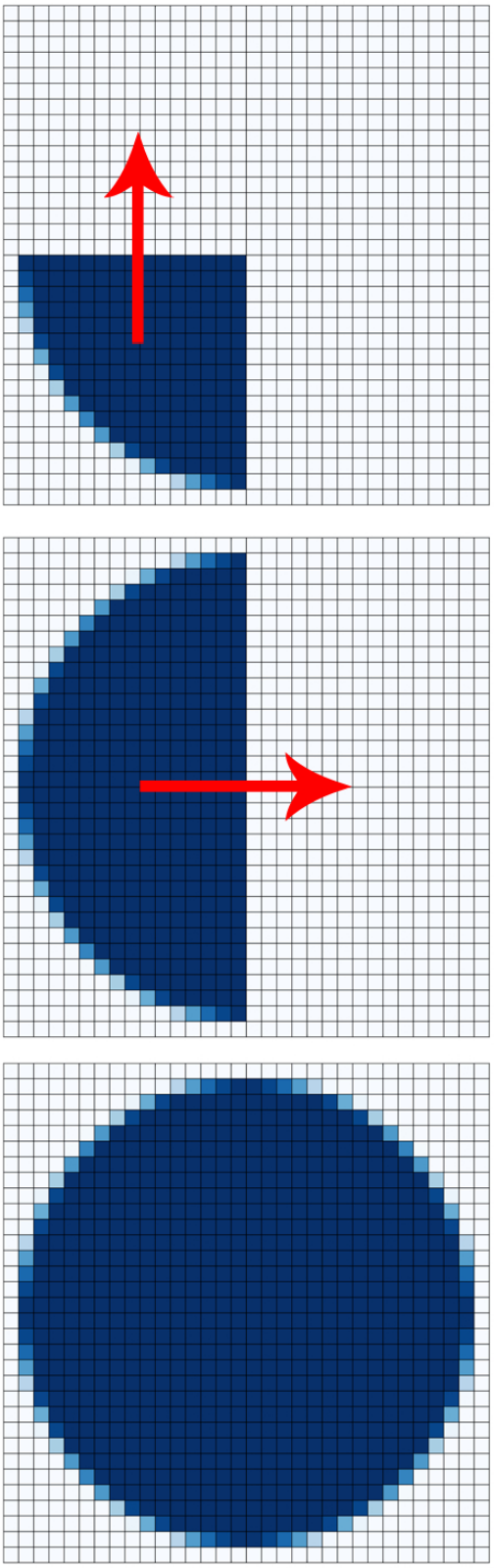


FIGURE 5.8: Graphical representation of the production of the kernel with minimal computational calculations by utilizing symmetry.

summation is computed at least $1,000,000$ times. For smaller sources with a kernel of 10×10 pixels, the convolution of a whole magnification map takes less than 1 second. Convolution of the same map using a larger source and kernel (100×100) can take over 30 seconds, with the time increasing proportionally to the source size. To reduce the number of calculations performed for large sources, a reduction in the resolution of the magnification map is made. Should the diameter of the source be larger than 50 times a pixel width, then a magnification map with a lower resolution is computed and used in the convolution process. A decrease in resolution of the convolution map will be less accurate than a map of higher resolution, but from experimentation it was found that this loss is negligible. It is acceptable to reduce the resolution of the magnification map with large sources for a significant reduction in the time, resulting in the convolution process always being performed in under 1 second.

Loop unrolling does not work for dynamically sized loops, so cannot always be applied to the convolution process. Alternate optimizations are achieved in the same way as previous maps, due to symmetry; only half of the map needs to be computed and then mirrored. In situations where the kernel size in known and takes longer than average to process, a special implementation of the convolution code exists that has fixed dimensions and therefore uses loop unrolling. This case is only valid for a convolution with a kernel of exactly the same dimensions (13×13). A flow chart in Figure G.4 shows the steps involved in performing the convolution process.

5.5.1.7 Light curve generation

Producing a convolved magnification map only requires three parameters d, q, and ρ. From a convolved map, a light curve of any trajectory can be produced using a single CUDA block. Given a set of data points with fixed times (t_i), a single thread computes a single data point. From the model parameters u_0, ϕ, t_0, and t_E, each thread calculates its data point's coordinates on the magnification map in units of Einstein radius (u_1, u_2), which represent where the centre of the source would be at time t_i. As long as the time scale is not too large, each of these coordinates fall inside the magnification map dimensions, from which the magnification can be interpolated.

To interpolate the magnification, each thread performs a single access to texture memory using the hardware accelerated bilinear interpolation at its u_1, u_2 coordinates. Although bilinear interpolation is very fast, it is not as smooth as bicubic interpolation. An optimized implementation of bicubic interpolation has been trialled, but due to the very sharp magnification gradient next to caustics, it was found that the bicubic interpolation had a tendency to smooth the sharp magnification changes too much. It also requires

additional calculations, making it a slower process than hardware interpolation. Therefore it was decided that bilinear interpolation would be more suitable. Should a data point represent a coordinate outside the magnification map dimensions, a single lens approximation at a distance (u_0) from the centre of mass is used. When appropriately large maps encompass the whole caustic structure and surrounding area, using a single lens approximation for points outside of these dimensions is acceptable as negligible difference exists between the models.

Assigning a single thread to each data point (or group of points) allows all magnifications to be determined in parallel. This makes the process rapid, as there are no requirements for any calculation to wait upon the outcome of any other.

Providing the same binary lens model parameters to every thread but unique time values makes it possible to build a complete light curve showing the change in magnification against time. Using this method, it is possible to compare how well any model fits a set of data by standard χ^2 methods.

5.6 Parameter space exploration

5.6.1 Initial grid search

The method outlined above details a new approach used to determine the light curve for a single set of binary lens parameters, which can be compared to a data set to see how well it fits. By testing different parameter sets and comparing how well they fit a data set suggests which parameter set is most likely to be the true set of parameters that produced the original data. This method can be used to determine areas in parameter space where the global minimum is likely to be. A grid search approach in u_0, ϕ, for each d, q magnification map, where t_0, t_E and ρ are minimized by simplex methods is used. This is repeated on a grid of d, q values, to produce a minimized d, q χ^2 map, from which areas where the global minimum is likely to lie can be found. To improve the accuracy of the model parameter solution, a MCMC method is used to explore local parameter space further and determine the global minimum parameter set. When the limb-darkening coefficient (Γ) is not a free parameter to explore, it is fixed at a value of 0.53 (in the I band). This coefficient is approximately that of a typical dwarf and giant star with an effective temperature 5500K from the tables of Claret (2000).

5.6.1.1 Grid search optimizations

To fit a model to data accurately, a complete search of all parameter space is required. Potentially this can be very slow, as it requires the process outlined above to be performed millions or even billions of times. To enable all parameter space to be explored in a reasonable length of time, ways to reduce the number of calculations performed, or the time to process each parameter set are essential. The use of a GPU enables unique ways of optimizing processes that are typically unfeasible on a standard PC. Therefore our new method has been developed so that the parallel capabilities of a CUDA device are fully utilized.

5.6.1.2 Magnification maps in a grid search

When optimizing a routine, the first step is to identify the slowest part and improve its speed or reduce the frequency that it is required, and ideally, both. For a single parameter fit, the slowest part of the method is the production of a magnification map. As this step takes the majority of the time, optimizations such as only computing half of the map have already been detailed, however, the vast number of complex calculations still required takes a significant time. Therefore reducing how often a map is made is crucial in preventing the full parameter space search taking an unfeasible length of time. As a magnification map only depends upon a unique pair of mass ratio (d) and separation (q) parameters, it is possible to search all other dimensions of parameter space for a single d, q pair before another magnification map is made.

5.6.1.3 Processes used to explore ρ

From a single magnification map a range of source sizes (ρ) can also be searched using a variety of convolution kernels. As different convolved maps are computed for each ρ used, there is a limit to how densely it can be sampled within a feasible time. To avoid producing a very large number of convolution kernels or searching a sparse ρ parameter space, a set of seven convolution maps are produced over a logarithmic range of ρ values, from 0.001 to 1.0. To further explore this parameter space, linear interpolation between the convolved maps is performed. This allows for a wide range of ρ values to be explored without the time spent producing numerous convolved maps. Although interpolation between maps will not produce accurate convolved magnifications for all values of ρ, it does provide a χ^2 gradient which can be explored using simplex downhill methods. The logic process involved in producing this set of convolved maps can be seen in the flow chart in Figure G.5.

5.6.1.4 Method to reduce map making calculations

The construction of a convolved magnification map array requires a very large number of calculations, so an alternate methods was developed to save time. Instead of recomputing an array of maps each time a grid search is performed, the arrays were stored on a local hard disk drive so that they could be read out of memory whenever they were required. This method was found to be slower than reproducing the maps each time. From a library of all the convolved maps needed to perform a complete grid search, a single memory access to retrieve the required maps takes over 14 seconds, where as reproduction of the same maps took less than 5 seconds. This does not rule out the possibilities of using a library of maps, as technologies such as solid state drives could be implemented to reduce memory access times, and possibly achieve greater performance.

5.6.1.5 Parameter search strategy

With a set of convolved magnification maps, the remaining parameters to explore describe the source trajectory (u_0, ϕ) and timing (t_0, t_E). A single source trajectory can be interpolated rapidly using the method outlined in section 5.5.1.7, but all possible trajectories encompass a very large parameter space. These remaining parameters are all very sensitive, so a small change in any of these can cause very large variations in the model. Therefore a small step size in their exploration is desirable.

The performance of interpolation is high enough that the slowest part of a single trajectory interpolation routine is the data transfer between the host and the CUDA device (i.e. storing a magnification map on the GPU). By reducing the number of times the routine has to go between the two devices, it reduces the computation time. Therefore a search strategy is used to explore all possible parameter space in a single magnification map (i.e. d, q pair), before a new map is created and searched.

By appropriately dividing the number of available threads into blocks within a grid, searching multiple dimensions of parameter space in parallel without multiple calls to the same function becomes accessible, which reduces the number of memory transfers of magnification maps. Each data point is assigned to an individual thread, so it can be inferred that a data set is contained within a single block. Each block can be assigned a unique parameter set which makes up a grid. The grid can be up to three dimensions and therefore enables three dimensions of parameter space to be explored, but there exist five unexplored parameters (u_0, ϕ, ρ, t_0, t_E) per d, q magnification map. Understanding how these parameters affect the trajectory and the light curve, help in developing a way to solve this problem.

Both u_0 and ϕ change the source path across the magnification map, which changes the shape of the light curve (e.g. the number of peaks). t_0 and t_E change the time of crossing and therefore the scaling of the light curve in time, and ρ changes the convolved map to be interpolated from, affecting the width and curvature of peaks. How these parameters affect the light curve shows that for each combination of u_0 and ϕ, the remaining three parameters (t_0, t_E, and ρ) can be more easily minimized together, rather than all five parameters minimized simultaneously. A search approach that uses this concept is performed by using a simplex downhill to minimize t_0, t_E and ρ for a fixed u_0 and ϕ combination.

5.6.1.6 Reducing parameter space with prior information

To reduce the size of the search space required to be explored, information based on the event's data is provided to the modeller. For events where the data follows closely to a Paczyński curve with no major deviations, an estimate for t_0 and t_E can be based on a visual inspection of the raw data. From these inputs, starting point coordinates for a three parameter simplex (t_0, t_E and ρ) are determined.

If the data does not follow the Paczyński curve model, the number of peaks in the data as well as the JD of at least one peak (preferably two) is provided. Each u_0, ϕ pair (with a point source ρ) has a distinct light curve shape whose number of peaks and locations (with respect to each other) do no change, it is possible to align the peak in the data with that of the model. A list has been created detailing for each d, q, u_0, ϕ combination the number of peaks and their locations in units of Einstein radii. This list makes it possible to align the data with the model where the following equation must be true,

$$t_0 = t_p - M_p t_E \ , \tag{5.12}$$

where t_p is the input time of peak from the data (JD-2450000) and M_p is the model time of peak in units of Einstein radii. However, not knowing either t_0 or t_E means a search over these two parameters is required. As t_E is known to usually be between a couple of days and a couple of hundred days, with the shorter timescale events being more favored (Sumi et al., 2011), a grid search spaced logarithmically in t_E is performed solving for t_0 at each point using Equation 5.12. The χ^2 at each grid point is calculated and the best solution is used to determine the starting points of the three parameter simplex.

If a model has multiple peaks with only one data peak time provided, the grid search is performed with the data peak aligned to each of the model peaks, with the overall best

solution from all peaks being chosen. When the input number of peaks is larger than one, a quick test is made to determine if the model contains enough peaks to match the data. If not, no further modelling of this u_0, ϕ is performed as it cannot be a viable solution.

The optimum situation occurs when multiple peaks and times are provided, enabling t_0 and t_E to be solved analytically for each viable trajectory. Because the locations of the peaks are known in units of Einstein radii for the model and as well as in JD for the real data, it is possible to solve for t_E as follows,

$$t_E = \frac{t_{p2} - t_{p1}}{M_{p2} - M_{p1}} \ , \tag{5.13}$$

where t_{pi} is the input time of the data's peak i, similarly M_{pi} is the model's time of peak i in units of Einstein radii. In both cases the peaks are ordered with the earliest peak occurring first (i.e. the JD of peak 1 is always smaller than peak 2). A χ^2 calculation is made for every possible combination of the data peaks aligned to each of the model peaks, taking the best solution for use in the distribution of starting points of the simplex.

5.6.1.7 Simplex downhill search strategy

The CUDA routine utilizes a simplex downhill approach originally developed by Nelder & Mead (1965) and more recently translated into C code (AMOEBA) for computational use in Numerical Recipes (Press et al., 1992b). This routine has been modified to remove Numerical Recipes' specific functions and to run in parallel on a multi threaded CUDA device. It is hard to estimate a source size from visual inspection of the data, therefore a small source value (close to the average value of $\rho = 0.003$) with a small variation in the simplex starting points are chosen. As this is a fairly small source size, it ensures that sharp caustic features are not smoothed out, and by using a simplex it allows for the movement to larger source sizes if appropriate. The amount of change to the model with respect to t_0 is dependant on the size of t_E, so the simplex starting points distribution of t_0 values are scaled based on the current best estimate of t_E. A simplex downhill approach has several advantages, in that it can quickly find the minimum of several parameters without the need to explore all the surrounding parameter space and it can do so more accurately than a grid search, unless a very high resolution search is made (which is very time consuming). However, it is not without fault, as the simplex only follows the downhill gradient of χ^2 space, there is a chance of the movement falling into a local minima and missing the global solution. This is a contributing reason as to why t_0 and t_E are solved by simplex downhill, whereas, u_0 and ϕ are searched using a grid

due to the nature of how they affect the light curve. Parameters u_0 and ϕ are much more likely to have several local minimum compared with t_0, t_E and ρ, they are also much harder to make global minimum estimates for as they can have very different local minima depending on the d, q combination.

5.6.1.8 CUDA grid search strategy

With t_0, t_E, and ρ found using simplex methods, only u_0 and ϕ are left to search per d, q map, meaning they can become the two dimensions in the CUDA grid. This completes a search of the five remaining parameters (u_0, ϕ, ρ, t_0, t_E) from a single call to the CUDA device (one d, q pair), which results in removing a significant portion of time otherwise spent transferring data to and from the GPU memory caused by repetitive CUDA device function calls.

Performing a grid search on a single d, q pair requires the call to the GPU to include the inputs of an array of convolved magnification maps (into texture memory) and a grid of u_0 and ϕ to start simplexes at. Upon completion it returns two dimensional arrays of minimized χ^2, t_0, t_E and ρ solutions, all of the same size as the input u_0, ϕ arrays. From the minimized χ^2 map it is possible to locate the co-ordinates of the best solution which are used to determine the associated parameter values from each of the other returned arrays. This results in a set of u_0, ϕ, ρ, t_0 and t_E parameters which produce the best solution of a single d, q pair. Repeating this process for each d, q pair of a grid search space builds up a minimized χ^2 map, from which it is possible to identify areas of local minima and ultimately the likelihood of a planetary detection.

A complete search over all parameter space is performed using the above method, with the resolution of the search being easily controlled by defining the step size in each parameter. For each set of parameters explored a χ^2 is determined on the GPU with each thread calculating the squared separation between the model and data for a single data point and storing it in shared memory. Each data point's contribution to the χ^2 value is then parallel summed across all data using the technique detailed in section 4.4.2.

5.6.1.9 Reducing trajectory search space

Many parameter combinations in a wide grid search will be far from the true solution yet still need to be explored, potentially adding a lot of wasted computation time. Therefore, a method to determine when sets of model parameters are not suitable is implemented. The raw data provides a limited amount of information about the model that fits it. As

previously mentioned the t_0 and t_E estimates can be read from the raw data and so can a lower limit on the maximum magnification of the event. Examining the most magnified peak in the raw data and assuming zero blending, it is possible to take the magnification at this point as a conservative lower limit of the maximum magnification. This requires a minimal awareness of binary lens light curve shapes to avoid considering possible outliers. However, knowing this additional piece of information is crucial in minimizing excessive model calculations. Despite every convolved magnification map having a trajectory that can produce a near infinite magnification, the majority of trajectories will achieve a maximum magnification much lower than this. It is possible to test every trajectory's maximum possible magnification against the estimate of the lower limit, if it falls short then the proposed model parameters should not be processed. The larger the lower limit of the maximum magnification value, the fewer trajectories that meet this requirement and need to be modeled. This resulting in a faster processing time for high magnification events. Implementation of this method shows dramatic performance gains.

On the u_0 and ϕ minimized χ^2 grid search, areas of local minima are relatively small compared with the whole parameter space explored. So an appropriately dense search space is required to ensure no minima are missed, however, this causes a large number of excessive trajectories that will not be near any local minima. To reduce the parameter space searched far from the global minimum, a method to explore only trajectories that pass close to caustic structures is used. A familiarity with caustic structures was developed, which lead to the development of an algorithm that determines the magnification map coordinate(s) that lie close to the centre(s) of the caustic shape(s), and a suitable radius around the coordinate(s) that ensures a whole caustic structure is enclosed within a circle. This algorithm provides a set of values that describe a reduced parameter space which encompasses all trajectories where possible solutions may lie for a given convolved magnification map (see Figures 5.9-5.21 for a graphical representation of the reduced search spaces). Not only does this method reduce the space explored, but it also provides the benefit of describing only areas close to the caustics irrespective of their size. This ensures that areas around small planetary caustics far from the origin are not missed, which can easily happen. If a search in ϕ is only performed around the origin, a distant small caustic would be very sensitive to a precise ϕ value, meaning the search resolution would have to be very high. By searching areas centred on caustic structures, this problem no longer exists, allowing a lower ϕ search resolution to be used yet ensuring no caustics are overlooked. Starting from the centre of each circle the algorithm produces, it is possible to explore a range of ϕ values from $0 - 360°$, and a fixed number of linear steps in u_0 from the centre out to its radius. This produces an even sampling over caustic structures and forces denser sampling of u_0 on small caustics. The outline of this algorithm is detailed as follows:

Calculation : Solve the forth order polynomial for the four caustic positions at 0° and sort the solutions in order of closest to furthest in y-displacement.

Situation 1 : Test if two points lie on the horizontal axis. If not go to Situation 4, otherwise:

1.1 Define three separate circular objects which encompass the whole caustic. The caustic structure may consist of up to three separate shapes, shown in Figures 5.9 - 5.12.

1.2 To determine the first circular object, take the average of the two points on the horizontal axis, and set this as a new origin.

1.3 The object's radius is determined by 1.75 times the absolute difference between these two points.

1.4 Solve the forth order polynomial for the four caustic positions at 90°, 180° and 270° and sort them in order of closest to furthest in y-displacement.

1.5 To determine the remaining two off axis circular objects x-axis coordinates, consider the 3rd solutions at 0°, 90°, 180° and 270°, taking the average of the x-axis coordinates at 0° and the solution that is furthest away. To determine the other circular objects origin repeat this process with the 4th solutions.

1.6 For the two off axis circular objects y-axis origin coordinates, take the average of the pair of absolute values used to determine the x-axis solutions, multiplied by the sign of the 0° solutions y-axis coordinate.

1.7 The radius of these two circular objects are 2 times the separation of the solution coordinates that were averaged.

1.8 Test whether the off-axis origins lie within 75% of the radius from the on-axis origin (a check for overlapping areas with the central area).

1.9 If a single off-axis origin lies within this range (not overlapping as much as the other circular object), equate the other circles origin and radius to be the same but mirrored across the x-axis.

1.10 If both lie within this range (both have a small overlap), match the radius and origins to the larger of the two objects (mirrored across the x-axis)

1.11 Similarly, if neither lie within this range (both have a large overlap), equate their coordinates and radius to be the same as the smaller of the two circular objects.

1.12 To ensure the search spaces do not overlap too much, determine the separation distance between the on and off axis origins.

1.13 If the radius of the on axis origin is greater than this separation, gradually decrease it until it is less than or equal to the separation, or 75% of the x-axis separation of the two on axis 0° caustic solutions.

1.14 If the off-axis radius is larger than this separation, similarly decrease it until it is less than the separation or 70% of its original radius.

Situation 2 : Test if the vertical separation of the two non-zero y-axis solutions at 0° are less than half the horizontal separation of the zero y-axis solutions, and if the two non-zero y-axis solutions in the horizontal axis lie between the x-axis coordinates of the zero y-axis solutions. If not go to Situation 3, otherwise:

2.1 Define two separate circular objects which encompass the whole caustic structure. The caustic structure is wide and narrow, as shown in Figure 5.13 - 5.15.

2.2 Sort the 0° caustic solutions by numerical order of their x-axis values.

2.3 Set the first circle's origin as a quarter of the way between the first two points in the x-axis, and a y-axis value of 0.

2.4 Similarly, set the second circle's origin as three quarters of the way between the third and forth points on the x-axis, and a y-axis value of 0.

2.5 The radius for these two origins are determined to be the larger of either, 1.2 times the x-axis separation between the two 0° solutions either side of its origin, or equal to the vertical seperation of the third and forth 0° solutions.

2.6 Following the same logic as 1.12-1.14 the two radii are checked and modified to avoid excessive overlapping.

Situation 3 : Test if the radius of the initial on-axis solution is greater than 75% of the separation between the on and off-axis origins plus their radius. If so:

3.1 Define a single circular object to encompass the whole caustic structure. The caustic structure is a single central caustic, shown in Figures 5.16 - 5.18

3.2 Sort the 0° caustic solutions into numerical order by their x-axis values.

3.3 Set the only origin point as the mid way point between the first and last of the sorted 0° solutions.

3.4 The radius is defined as the 1.4 times the distance between these two points.

Situation 4 : Test if all four of the 0° solutions lie on the horizontal axis. If so:

4.1 Define two separate circular objects which encompass the whole caustic structure. The caustic structure is wide and contains two separate shapes, shown in Figures 5.19 - 5.21

4.2 Sort the 0° caustic solutions into numerical order by their x-axis values.

4.3 Set the first origin as a quarter of the way between the first two points in the x-axis and a y-axis value of 0.

4.4 Similarly, set the second origin as three quarters of the way between the third and forth points on the x-axis, and a y value of 0.

4.5 The radius for these two origins are calculated as the separation distance between the two 0° solutions either side of their origin.

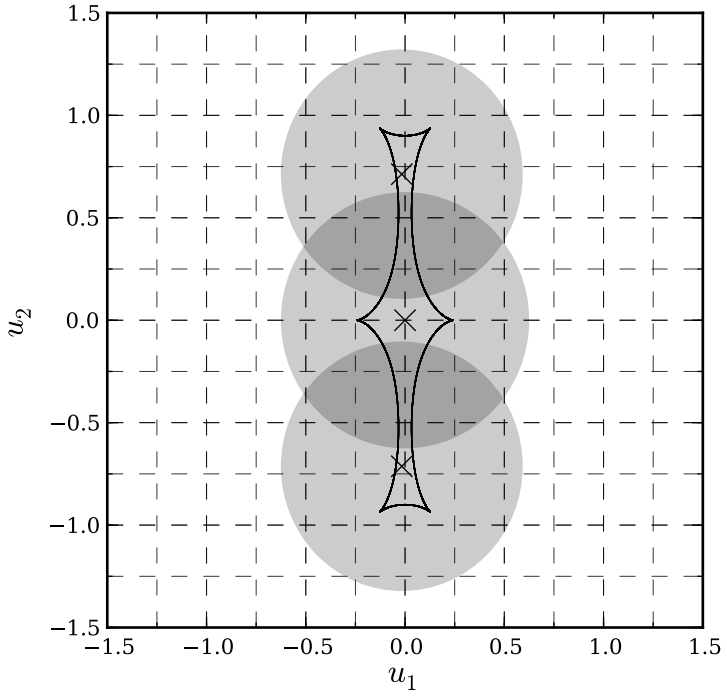


FIGURE 5.9: Example of three search spaces encompassing a single tall thin central caustic.

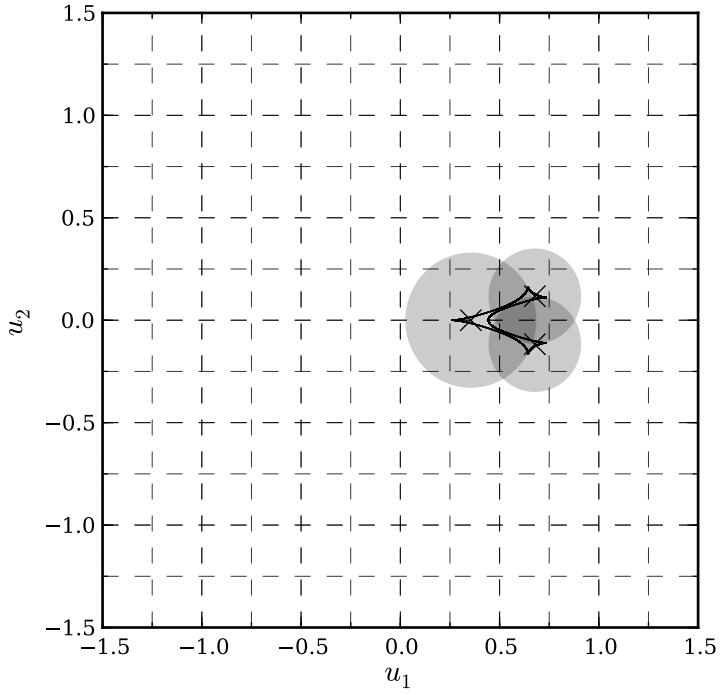


FIGURE 5.10: Example of three search spaces encompassing a single skewed caustic about to split into three separate caustic shapes.

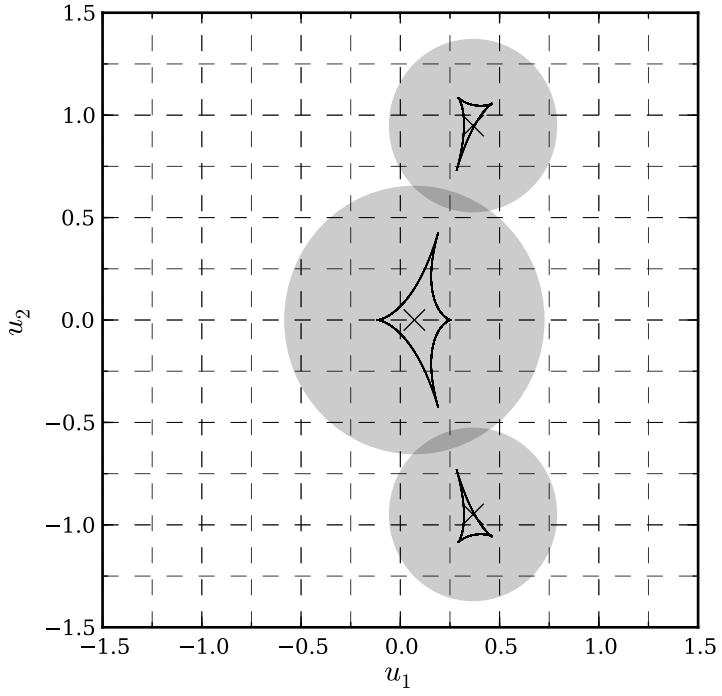


FIGURE 5.11: Example of three search spaces encompassing a three large recently split caustic shapes.

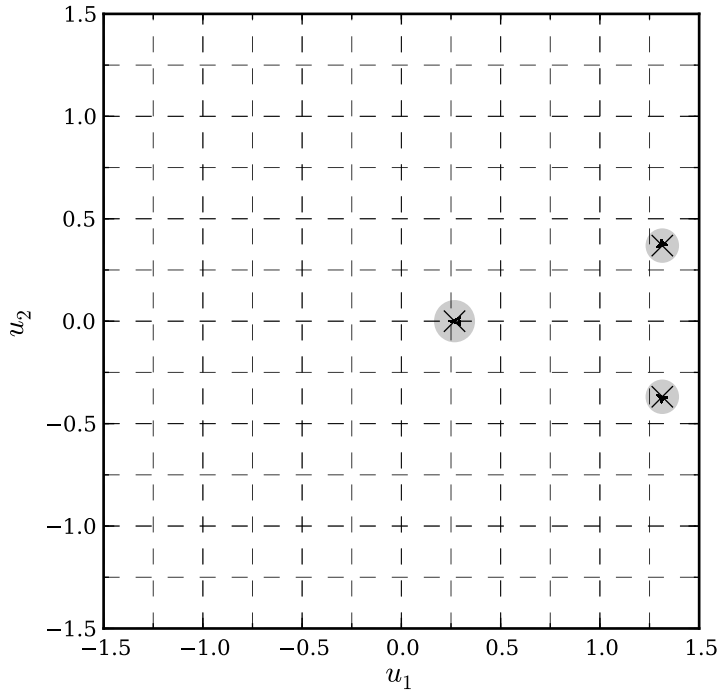


FIGURE 5.12: Example of three search spaces encompassing three widely separated small caustics.

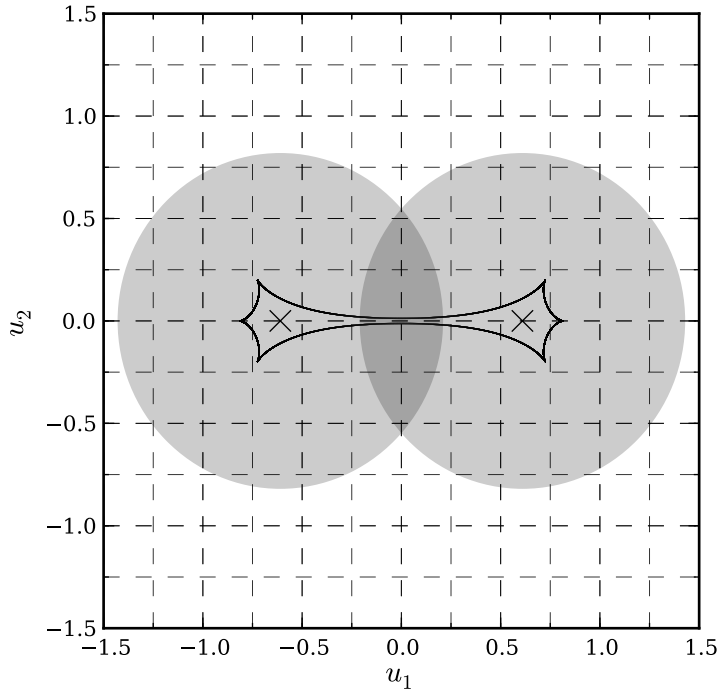


FIGURE 5.13: Example of two search spaces for a single symmetrical wide caustic.

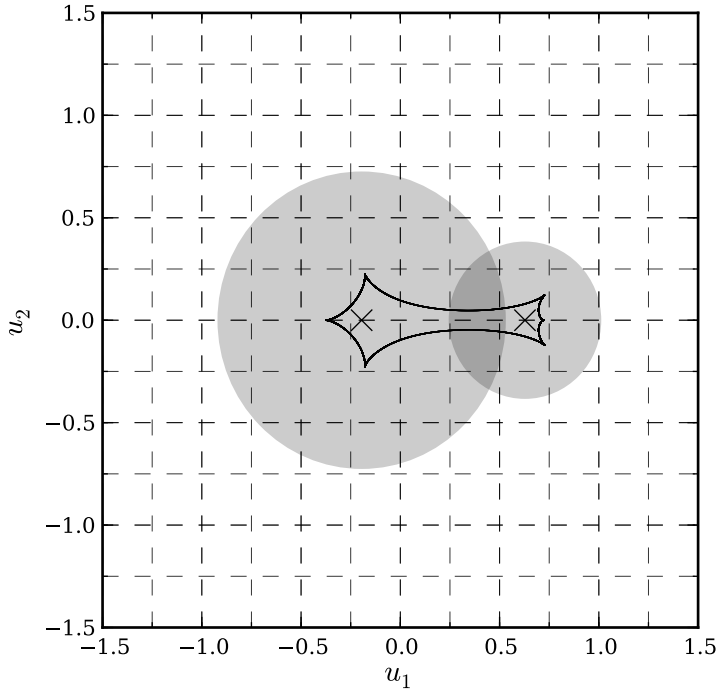


FIGURE 5.14: Example of two search spaces of different sizes for a single skewed wide caustic.

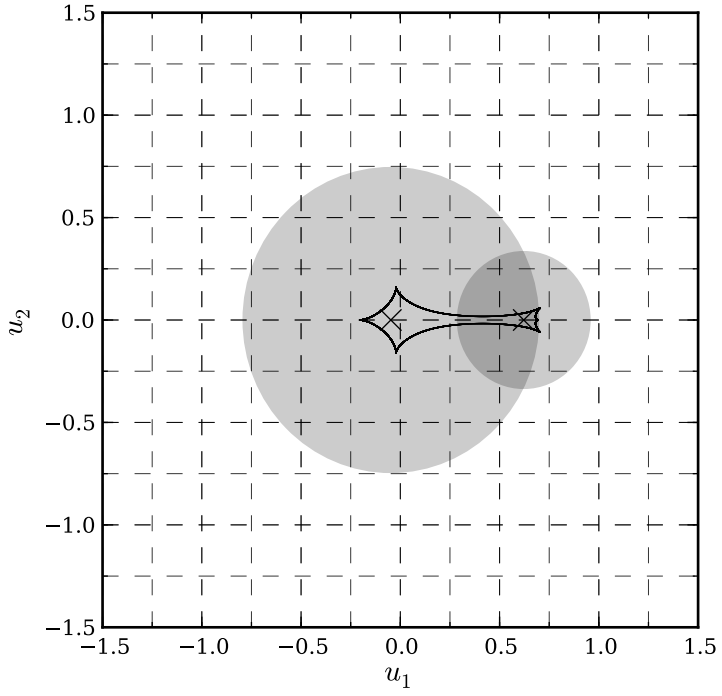


FIGURE 5.15: Example of two search spaces of different sizes for a single skewed caustic that is about to split into two separate caustics.

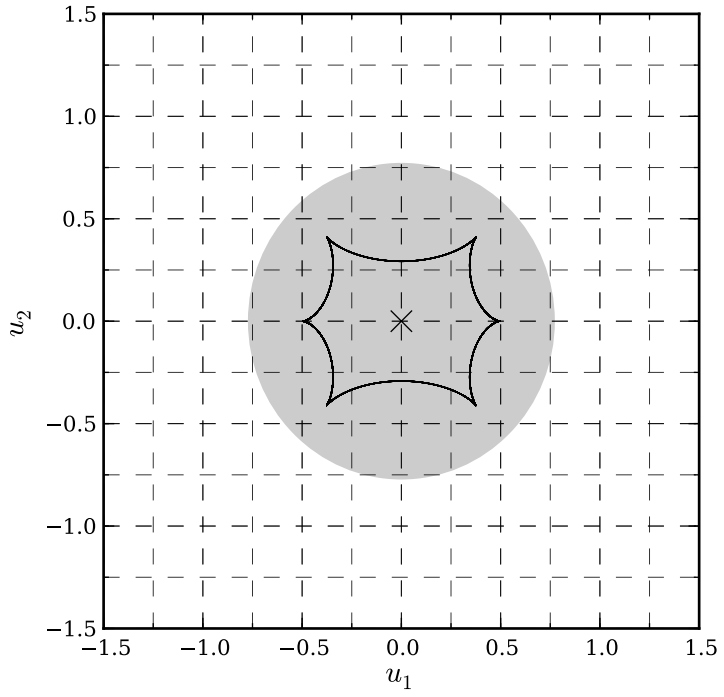


FIGURE 5.16: Example of a single search space encompassing a single central caustic.

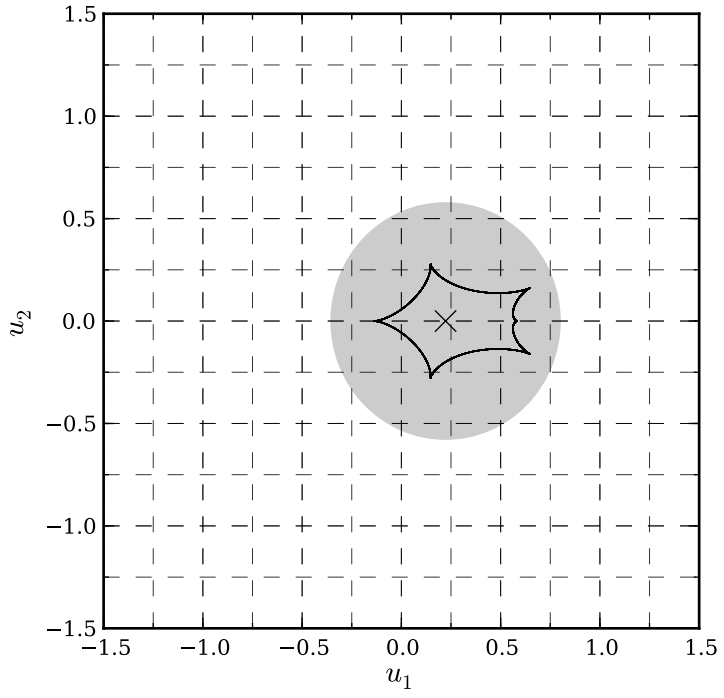


FIGURE 5.17: Example of a single search spaces encompassing a skewed central caustic.

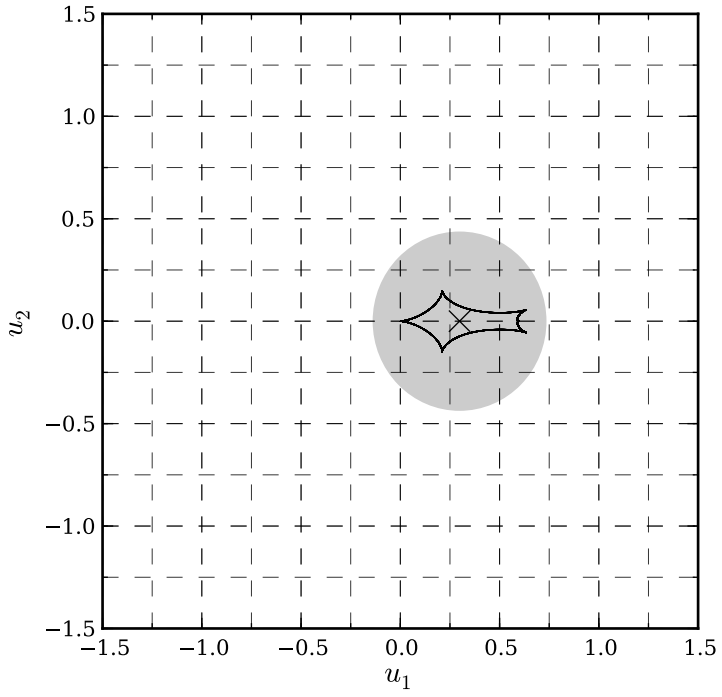


FIGURE 5.18: Example of a single search space encompassing a wide, skewed single central caustic just before it requires two search origins.

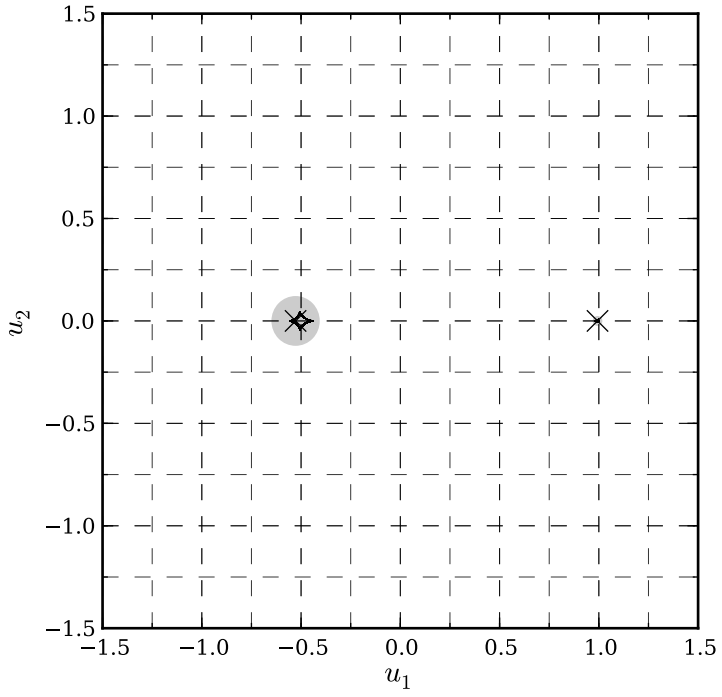


FIGURE 5.19: Example of two search spaces of different sizes with a large separation.

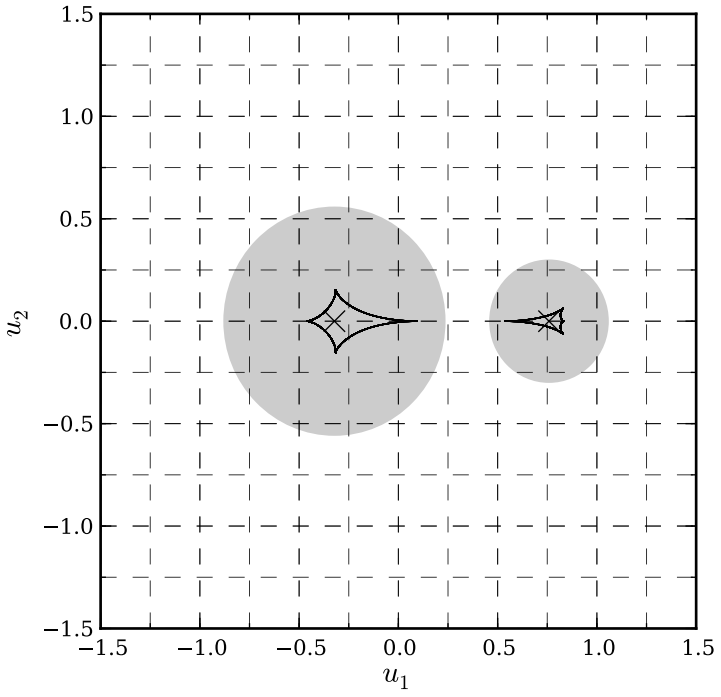


FIGURE 5.20: Example of two search spaces of different sizes with a close separation.

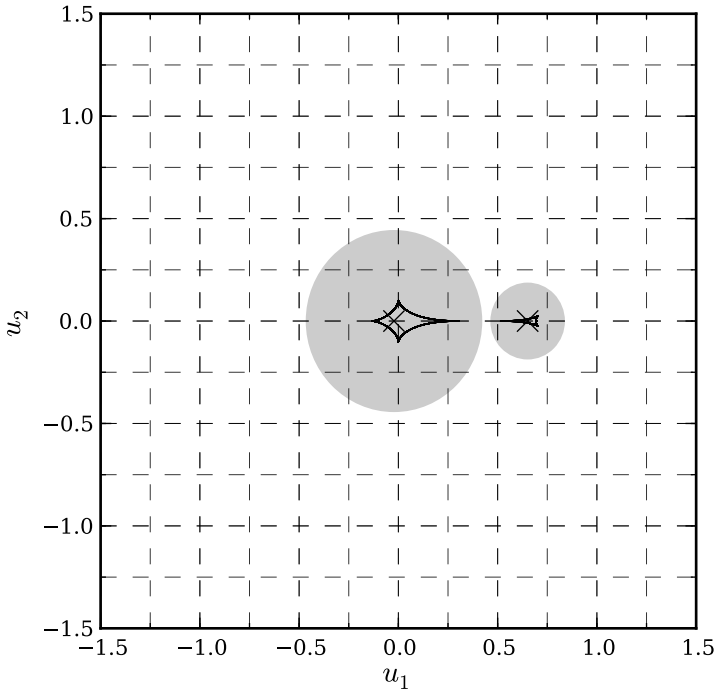


FIGURE 5.21: Example of two search spaces of different sizes that are about to merge.

5.6.2 Local Markov Chain Monte Carlo

Upon completion of the seven parameter grid search that produces a d, q minimized χ^2 map, another process is initialized to help further explore parameter space to a finer accuracy. Some approximations are made to optimize the grid search modelling process, which leads to slight inaccuracies in the model. Together with using a grid search method, there is a limit to the accuracy of the parameter set that can be found. These inaccuracies could be improved by using a more accurate model and a higher resolution grid search. This is why the addition of a localized MCMC method enables the exploration at a greater resolution of the lowest χ^2 areas of parameter space, with a more accurate model. The model used in the MCMC process is based on the ICIRAS method, developed by Bennett & Rhie (1996), and detailed in section 2.5.4.3. The ICIRAS model is more accurate than interpolation, but it cannot be feasibly implemented into the initial grid search as its computation time to perform a full search is significantly worse than the rapid trajectory interpolation method previously detailed.

As the grid search has a fixed search resolution in d, q, u_0 and ϕ, the MCMC method helps explore the areas between the grid steps. This cannot be performed at every pixel of the d, q grid within a feasible time frame and to only consider the lowest pixel is undesirable, as the parameter space may have a global minimum in a narrow valley that falls between two pixels. In an attempt to avoid this, the best six pixels in the minimized d, q, χ^2 map are chosen to have MCMC searches started at. Forcing the best pixels to be an even mix of close and wide solutions (3 for $d < 1$ and for $d > 1$) attempts to ensure the known close/wide degeneracy is fully explored.

Implementing the ICIRAS modelling approach as an MCMC enables the search to explore surrounding local minimum areas of parameter space, with the ability to jump out of the minima to pursue the global solution. MCMC methods have some advantages over grid search techniques, and if left exploring parameter space for long enough they are guaranteed to always find the global minimum. However, knowing when they have achieved this or how long they will take, is not an exact science. No one method fits all solution for optimizing the search exists. Several trials have been made in an attempt to enable the chains to run long enough to locate the minimum without taking too long. This lead to the implementation of an MCMC search method detailed by Bennett (2010), which uses some of the MCMC methods detailed in section 5.2.3. Bennett's MCMC search method differs from the conventional approach by only conserving the most recent steps, with older steps being discarded. It is designed for the purpose of exploring the narrow valleys that twist and turn, winding through a multidimensional parameter space, ultimately leading to a minimum. Error estimates of the minimized

parameters are not determined by this method due to discarding most steps, so an alternate method is used to fine tune the accuracy of the minimum at the global minimum and determine the variances of the parameters.

To enable the MCMC process go as quickly as possible, the modelling process needs to be optimized as much as possible. Taking the method of Bennett (2010), an implementation has been developed on a CUDA device to rapidly model any binary lens parameter set. The logical paths that the MCMC process follows are shown in Figure G.8.

Six local MCMC chains are processed, and the chain with the lowest final χ^2 is determined to be in the area of global minimum, completing the search for the minimum. An outline of this search process is shown in Figure G.12.

5.6.3 CUDA image centred inverse ray shooting

The ICIRAS method can be processed in a single CUDA function call (hereafter known as CICIRAS) by suppling a complete binary lens parameter set (d, q, ρ, u_0, ϕ, t_0 and t_E), with a pair of arrays for the u_1, u_2 coordinates describing a set of caustic points. Using 128 threads and a one-dimensional block equal in size to the number of data points (Julian Dates) to be processed, enables multiple blocks to be processed simultaneously which produces a total magnification for each input data point.

5.6.3.1 Single lens approximation

Each block is a separate data point in the source trajectory which is defined by the time of observation. The first stage of the CICIRAS method is determining where the data point lies in the lens plane, its u_1 and u_2 coordinates. Avoiding excessive calculations is crucial in speeding up processing time, therefore before the full ICIRAS method is implemented a test is performed to determine if a point source approximation can be used. If the source is sufficiently far from the caustic structure(s) of the lens, a point source solution will suffice. Using all (128) threads, distance tests to all the recorded caustic points are made. If no thread finds the source to be too close to the caustic ($< 10\rho$), a single thread calculates the image positions and total magnification as if it is a point source, using the fast root solving routine developed by Skowron & Gould (2012).

5.6.3.2 Test for caustic crossings

The data that contributes to anomalies in events do not exist far from the caustics and are therefore solved for by an ICIRAS method. Using all 128 threads available, they are evenly distributed around the source edge, except one, which is at the source centre. The 5th order polynomial solutions for each thread are found using the faster root solving process. They determine the real solutions and magnifications at their coordinates and if they have five images, set a flag to identify that at least some of the source lies inside the caustic. As a finite number of points on the source edge are used, there is always a possibility that the area between the finite sampling may cross a caustic. To reduce the likelihood of this, the source point that has the largest magnification and is therefore likely to be closest to a caustic is used to create a new set of coordinates on the source edge. A new set of 127 points are evenly spread between the two previous points that were either side of the one with the highest magnification. As before the number of images are determined and if a single point lies inside the caustic, a flag is set. Oversampling of the highest magnification area should capture any caustic crossings that were previously missed, if any exist. In the unlikely event that the caustic crossing is still missed, its contribution to the total magnification from the small additional images it would produce are negligible, and therefore suitably ignored.

5.6.3.3 Determine how many separate images exist

From the above calculations the total number of images can be determined. If none of the points on the source edge or centre fall inside the caustic, then only three images exist. Should all of the source fall inside the caustic then 5 images exist. Alternatively if a point on the source edge but not the centre lies inside the caustic, five images exist where at least two of them are merged together. The latter situation requires special considerations. As the name ICIRAS suggests it is based on image centres, but from the source centre, there exists only three image centre solutions, when there are known to be five images. Therefore, the source edge solutions which fall inside the caustic are required to determine a 4^{th} coordinate inside the merged image.

5.6.3.4 Determine merged image centres

The point inside the merged image is found by a single thread. It first determines where on the source its edge crosses the caustic. This is found by counting how many images exist at each point on the source edge and recording when the number of images changes from three to five. Of the five solutions their separation distances are calculated, from

which the pair of images with opposite parity that are closest together are accepted as the two new image positions caused by entering the caustic. These image coordinates appear close to and either side of the critical curve and lie on the edge of the merged image.

To determine the area of an image, it is easier if the image centre is known. With merged images, to aid in the area calculation an estimate for the centre of the area of the merged images is determined. Starting at one of the new image positions on the merged image edge, a logarithmic distribution of all the threads is made in the shape of a cross. Each thread determines how far an inverse ray shot from its current position would land from the centre of the source. A single thread loops through each branch of the cross moving outwards from the centre, recording the thread which lands closest to the centre of the source. Only points in the cross which fall inside the original image are considered, this is achieved by ignoring the rest of the cross' branch being checked should any point falls outside of the image. This helps prevent the selection of a point in the cross which lies inside a closely neighboring image, an example of this process is shown in Figure 5.22. Upon completion a single point in the cross has been found which shoots a ray that lands closest to the source centre, therefore this point must lie closest to an image centre.

5.6.3.5 Image size estimates, creating image grids

The production of the grids that will encompass each image, start at the points closest to the image centres. A cross is set up (this time in polar coordinates) to make an initial calculation as to where the boundaries of each image are. Starting either in the radial or angular direction, each thread is assigned a coordinate logarithmically spaced from the central image position moving outwards. In the angular direction, the threads are spread between the limits of $\pm\pi$ and radially between $\pm 2.5\rho$. The threads are distributed using a twos complement method to determine if a thread is even or odd, so that it can be shifted up or down respectively. All threads perform inverse ray shooting to determine if they fall inside or outside of the image. This allows a single even and odd thread to count outwards from the centre until it reaches the first point that lies outside the image, which is declared as the image boundary.

Should a boundary be located within three points of the image centre, an additional step is made to help deal with the very small image and improve the accuracy of the boundary location. If this occurs the same process of shooting rays and checking if they fall inside or outside of the source is repeated in the current axis (angular or radial), however, the threads are now distributed linearly between the two boundary points previously found.

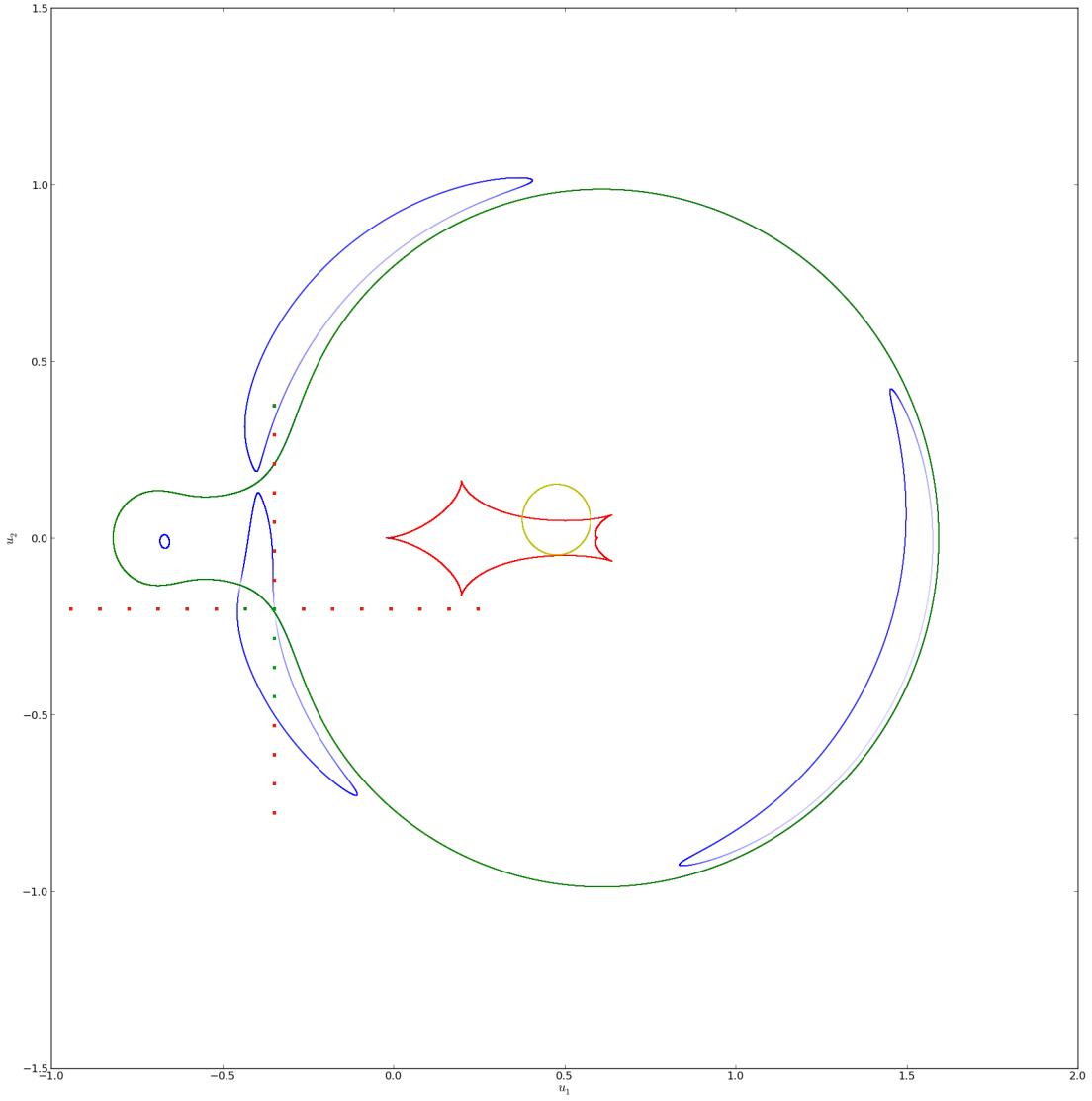


FIGURE 5.22: A set of four separate images (blue lines), passing close to and over the critical curves (green line), produced by a finite source (yellow line) crossing a caustic (red line). One image is crossing the critical curve and the centre of this image is unknown. A cross of points (dots, green inside an image, red outside any image) are sampled to determine which lies closest to the centre of the image. Green dots represent inside and image, and red outside of an image). This figure also includes an example of the cross overlapping multiple images showing the need for care when locating the centre of the correct image.

This step is repeated until the boundary is found more than two points away from the centre, which avoids creating a boundary that is at the same grid point as the image centre.

When locating the radial boundary, an additional consideration is required. Due to the grid being defined in polar coordinates, a negative radial distance is the same as its positive value, rotated by π. If a negative radius exists it could lead to a grid that would overlap itself, resulting in over counting the magnification of an image. To avoid this, an inner radius that is less than zero is prohibited. If the initial search locates the boundary to be less than zero, it is changed to a small value just larger than zero (10^{-6}). If the image really does encompass the origin (inner radius = 0), it will later be identified when the grid sizes are tested and resized.

5.6.3.6 Growing the image grids

Lensed images can have unusual shapes, as shown in Figure 5.23, or curvatures that do not closely follow the angular axis of polar coordinates. This means a grid using the dimensions previously found may not fully encompass the whole image. To ensure the whole image is enclosed, each grid is grown one edge at a time. Starting with one edge, every thread is uniformly distributed along its length and inverse ray shooting is performed to identify if any point falls inside the image. If so, the edge is moved outwards by a percentage (30%) of its current size and the process is repeated until every point along the edge is outside. This is repeated on each of the four edges. For each edge that is enlarged, the two adjacent edges need to be rechecked to confirm the previous growth has not affected the rest of the grid. A graphical representation of this process is shown in Figure 5.24.

A percentage growth is used to speed up the process by allowing bigger steps with larger images, however, for steps that are too large they may too easily jump across gaps into adjacent images, resulting in a merged image grid. While a merged image is not problematic, it does reduce the accuracy of the integration due to the use of a finite number of sample points within a grid (128×128), irrespective of image size. To help avoid this, an upper limit to the size of the percentage growth in the angular direction is set. No upper limit is set in the radial direction however, as this dimension is typically smaller than the angular size and will not result in large step sizes. Images do not closely overlap radially, so it is unlikely that a radial grid growth will accidentally merge two image grids together.

Special situations exist that require careful consideration when growing a grid, for example if an image makes a full ring, the angular size can grow forever. This is prevented by

using flags to signal a ring exists and stop the growth in each direction. Triggering the flag occurs when the angular growth is more than 2π, if both angular growths (clockwise and anti-clockwise) trigger the flag, then a full ring exists. The grid's angular edges are set to $\pm\pi$, and a new flag is set to identify that the image is a ring, enabling a different treatment when the integration is performed.

Radially a different consideration is required when working close to the origin, as it is possible that the grid may grow into negative radial space. If the radial edge proposed has a negative value, inverse ray shooting is performed at the origin ($r = 0$). If this test confirms that the image encloses the origin, the inner radius is set to zero. This means no inner radial boundary will exist, as the grid will be a full circle. However, if the origin is not enclosed, the inner radial edge is determined to greater accuracy. To locate the image boundary inverse ray shooting is used half way between the previous point which was inside the image and the proposed point which has a negative radius. This process of percentage growth when $r > 0$, and half a step back when $r < 0$, is repeated until a positive radial boundary is found where no points on the edge of the grid fall inside the source, as shown in Figure 5.25.

5.6.3.7 Image grid column integration

Grids that encompass full images are integrated to determine the area of the image enclosed. Using all available threads (128) enables a whole column to be integrated in one parallel step, this results in a rapid accurate area calculation per column. Because the image does not stay the same radial width along its angular length, it is possible that there are columns in the grid were the image exists in a very small fraction. To improve the accuracy of the integration the boundaries of the image within a column are determined to a greater accuracy. With an accurate boundary location, inverse ray shooting is performed in the reduced region including limb darkening.

To locate the image edge within a column each thread is assigned a unique coordinate on the grid, with the same angle but different radial values, they are evenly spread between the grid's inner and outer radial boundary limits. Inverse ray shooting is used to determine if each point lies inside or outside the image by returning a 1 or 0 respectively. As this calculation is performed to determine the images radial boundary to greater accuracy, limb-darkening is not included to improve computation performance. Initial identification of the boundary is performed by a single thread which counts inwards from the edge of the grid and locates the last point in the column before it enters the image for the first time. Working from the outer edge inwards is important to ensure that unusual shaped images do not have areas overlooked, as shown in Figure 5.23.

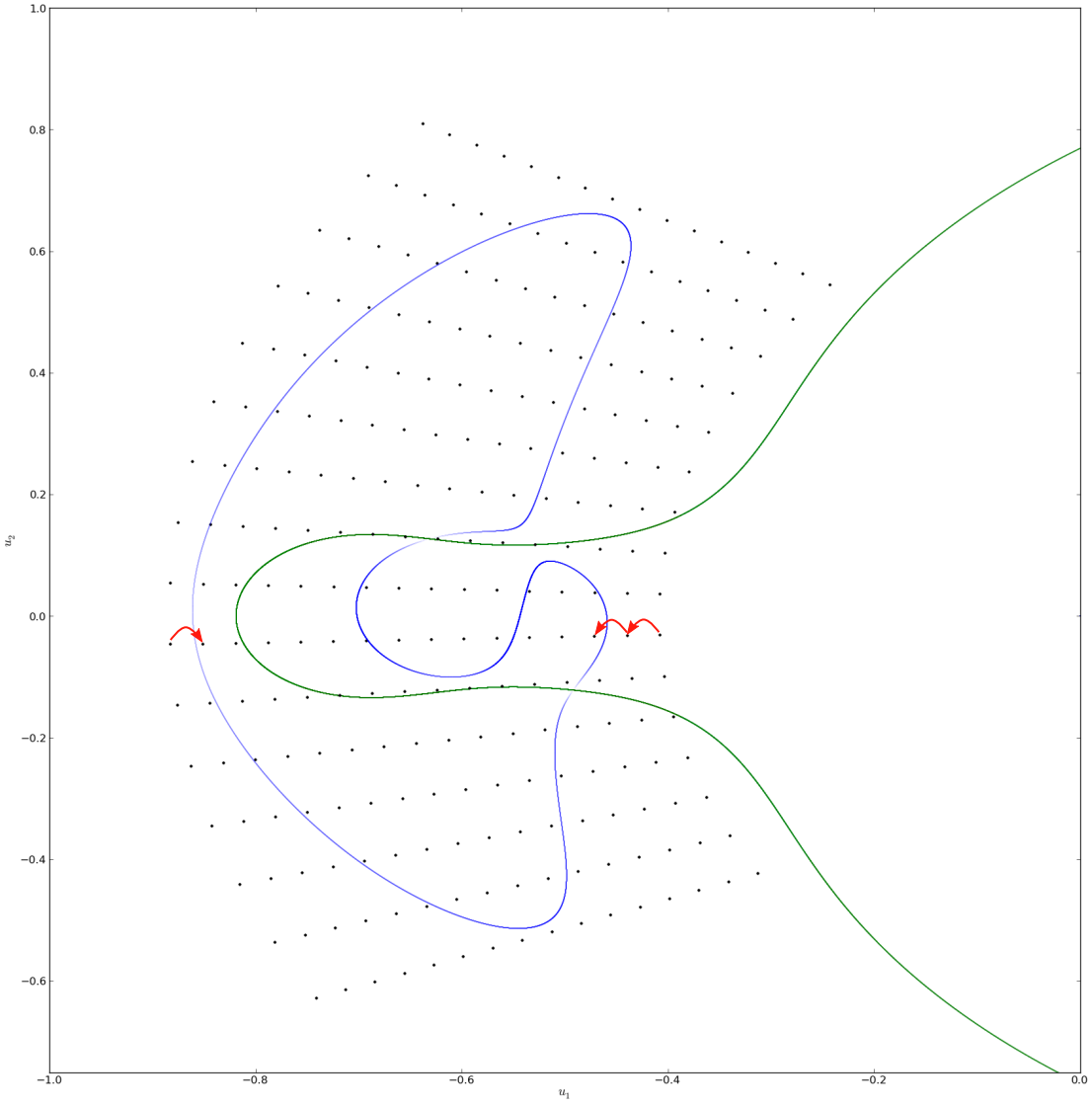


FIGURE 5.23: A microlensed image (blue line) of a finite source displaying an unusual structure as it passes over a critical curve (green) that can lead to complications in determining boundary edges. An example grid of points (black dots) are set to encompass the image, and the boundary location of each column needs to be found. The maximum width of the image is determined by working towards the centre from the outer edges (red arrows), locating the first point to fall inside image.

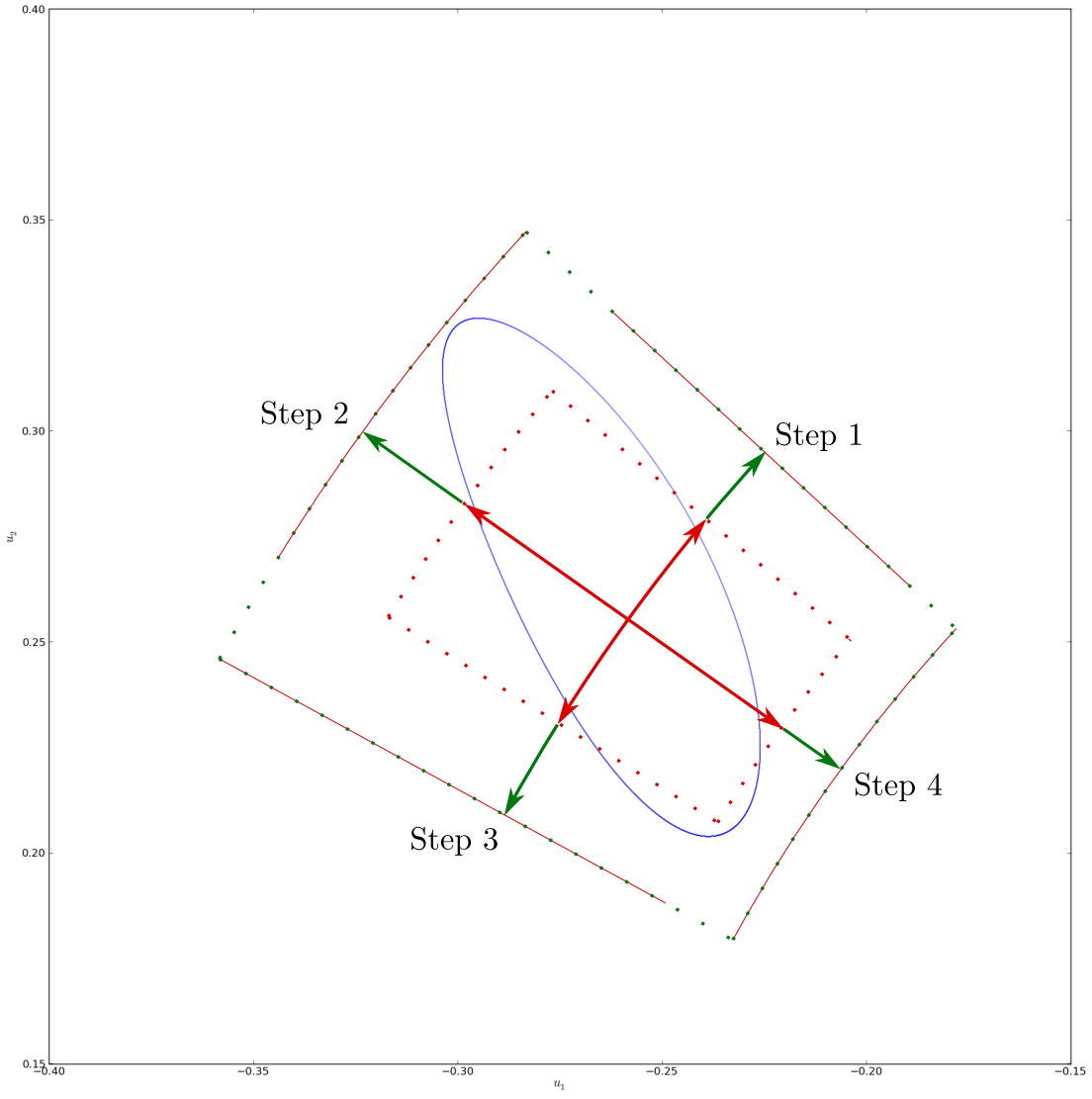


FIGURE 5.24: Setting up and growing the grid around an image (blue line) for image centered ray shooting. Starting from the image centre, the initial borders (red dots) are calculated by working outwards radially and angularly (red arrows). Each border is tested one at a time and grown outwards (green arrows up to the red line) until no point along its edge falls inside the image. The process is repeated until all borders are calculated to be outside of the image.

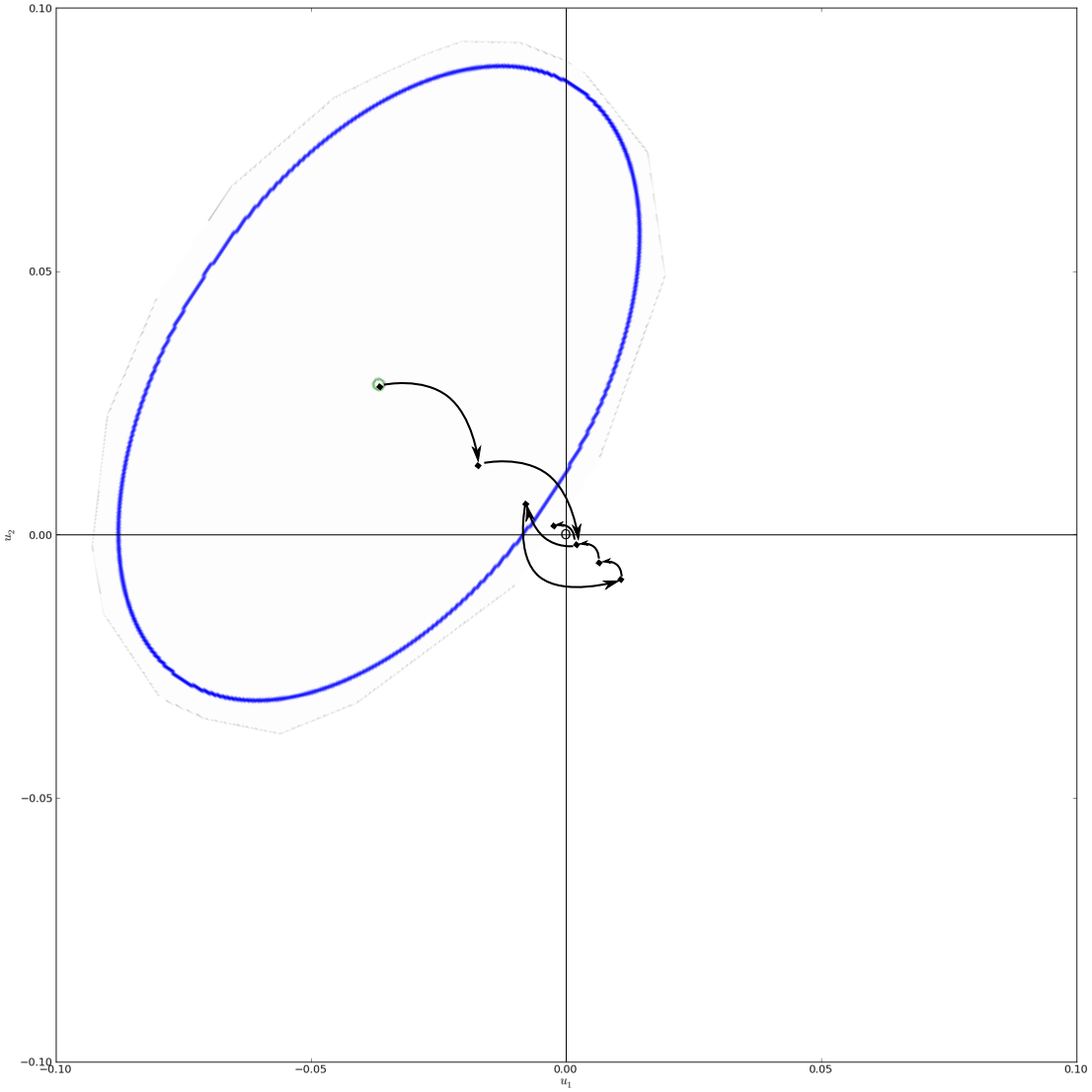


FIGURE 5.25: An example of the algorithm used to determine the inner radial edge of the grid boundary, ensuring no negative solution is chosen.

By repeating the inverse ray shooting method with all threads uniformly distributed between the two points either side of a calculated boundary location results in improved accuracy. To determine this boundary location, instead of a single thread counting through an array, the returned values from the inverse ray shooting can be parallel summed. Quickly providing an integer that corresponds to the thread number which is next to the boundary. Repeating this process means each repetition increases the accuracy of the boundary location by 1/Number of threads, until the desired accuracy of 10% of the grid's column resolution squared, or an accuracy greater than 1.0×10^{-8} is achieved.

Both (inner and outer) image boundaries are determined in this way per column, allowing a new evenly distributed set of points to be established between them. Inverse ray

shooting is performed on the new array, returning a limb darkened value. Integration of the limb-darkened column is calculated by summing together the array. A minor alteration is required to be applied to the summation as each point contributes an area of the image, but as the two end points of the column are located on the image boundaries, half of the area is outside the image hence their contribution is halved. Loop unrolling is implemented to sum the elements of the array and therefore determine the integral of the column, which is then multiplied by the column grid spacing. When the image grid is a full circle, there is no inner radial boundary. As the inner boundary does not need to be located, the search process is skipped and its radial location is set to zero. Every column in the grid is processed in this same way resulting in a row of integrals, which can then be integrated to determine the image's brightness.

5.6.3.8 Image grid row integration

Row integration requires the boundary to be located to a greater accuracy in the same way as detailed for columns. When determining the angular boundary, it is unlikely to be exactly half way between the inner and outer radius of the grid, therefore the radii of the image edges at each angular end of the grid are required. This leads to an additional step in the column integration being implemented, which identifies if it is the furthest clockwise or anti-clockwise column of the grid inside the image, if so the radius which is half way between the radially calculated boundaries of the column is stored. Angular boundaries are then found in the same way as the radial boundaries, with a single thread locating the boundaries from the initial array of integrals. Followed by every thread being distributed uniformly between the two points either side of the boundary locations and tested to locate the new image boundary. The process is repeated until the desired accuracy is achieved.

It is not feasible to resample between row boundaries as is the case with column integration, because it would require resampling all column integrals too. To account for the minor discrepancy between the actual boundary location and where the grid located it to be, a mathematical operation is used, Equation 2.15, where A_i is equal to $\left(\frac{9}{8} - \frac{\eta^2}{2}\right)$, B_i is $\left(\frac{3}{8} + \eta + \frac{\eta^2}{2}\right)$, and $\eta = (y_i - y_L)/h$ (where h is the row's sampled resolution). These two modifiers are applied to the two values at either end of the row, as detailed by Bennett (2010). Loop unrolling is used to perform the integration by rapidly summing together the integrals, determining the total brightness contribution from an image.

5.6.3.9 Check for double counting images

The CICIRAS process described is repeated for each image, with careful management to avoid double counting of image areas. Before any image is processed, it first checks to see if its central image position has already been enclosed within a previously computed grid, and if so it is skipped. Similarly, after each image calculation a test is performed to ensure its grid had not grown to encompass any previously calculated images, if this occurs the encompassed image grid is removed to avoid double counting its contribution.

5.6.3.10 Binary Lens EMCEE Search

Upon completion of local area MCMCs, the best solution is chosen from the chain that produced the lowest final χ^2 model, which is identified as the area of global minimum. From this solution the data errors are normalized by taking the square root of the χ^2 per degree of freedom (DoF) for each set of data (Equation 5.14),

$$\text{error scale} = \sqrt{\frac{\chi^2}{\text{DoF}}} \quad . \tag{5.14}$$

The normalized error data is used to prevent bias from different data sources in the modelling of an event when determining the global minimum and calculating the parameter set's standard deviation. To find the global solution to a greater accuracy, another MCMC process is used. As detailed earlier (section 5.2.4) a proven method, EMCEE, exists that successfully determines the minimum and standard deviation of a parameter set given a minimal input. Large numbers of tightly distributed parameter sets centred around the current best solution, and an error normalized data source are all that are required to initialize the EMCEE search. This process uses the CICIRAS modelling method for rapid, accurate model calculations of each proposed parameter set. Performing a burn-in and sample chain with EMCEE produces a well-converged solution that is analyzed in the same way as with a single lens event. Histograms of the parameter distributions are made to determine the mean and standard deviation of each parameter.

5.7 Summary

A new modeling minimization method for both single and binary lens events has been presented in this chapter. Our alternate modeling methods of using Bayesian inference

and a mixing model are detailed, explaining what they are and showing their use in a minimization search for single lens events with the python EMCEE minimization method. Upon starting a MCMC search the initial input parameters need to be carefully chosen, these choices are presented and justified along with the priors of each model parameter. From the result of an EMCEE search, the approach used to select the parameter set that achieved the largest likelihood value as the global solution is explained.

A different approach for solving the model minimization search of binary lens events is shown, achieving significant performance gains over a standard CPU method, with the use of a NVIDIA CUDA device. To produce a binary lens light curve, 5^{th} order polynomial solved magnification maps and ICIRAS methods are used. The implementation of these methods on a CUDA GPU device is thoroughly detailed, including the limitations found and optimizations implemented. Where possible time comparisons are presented to show the performance gains achieved when using a GPU device, as in the example of producing a point source magnification map which is over 1200 times faster on a CUDA GPU than a CPU. With the inclusion of finite source effects, modifications to the point source map are required. These processes are detailed including the optimized methods used to increase the resolution of the point source magnification map, and the routine applied to ensure no individual pixel distorts the map's magnification gradient. With a magnification map of high enough resolution a convolution kernel is used. An optimized method to produce a finite source kernel with limb-darkening is shown, which produces a whole kernel in ≈ 0.005 seconds, before it is applied to the point source magnification map. A convolution process is explained and the CUDA optimized method that has been developed is detailed, which ensures the convolution of any map size used does not take longer than 1 second. In a special situation which is frequently performed where the general case convolution process is known to contribute a noticeable amount of time to the whole convolution process, an alternate custom optimized routine has been developed to reduce the processing time further.

By the use of unique attributes on NVIDIA CUDA devices, an optimized method to interpolate the light curve from a magnification map is presented. Although interpolation of a convolved magnification map is only an approximation, the rapid results achieved made it a suitable method for use in the development of our new model minimization search strategy. Using magnification maps and interpolation, an initial large search of parameter space is achieved by a combination of grid search and simplex downhill methods. The search is optimized in numerous ways with the inclusion of input parameters such as the requirements of a maximum change in magnitude, and total number of peaks achieved. These inputs and search strategies enable a reduced parameter search space, to help minimize the time of the models initial search. Additional optimizations have been developed; such as ensuring only the parameter space around caustic structures

is explored. This not only avoids excessive time spent searching parameter sets far from any viable solution, but also ensures potential solutions around very small caustic structures are not missed.

Upon completion of a grid search, the further exploration of parameter space by use of MCMC is presented. This method ensures possible close, wide degenerate solutions have the chance of being explored, and enables the fine-tuning of all parameter values, which are no longer constrained by a grid search resolution. The use of magnification maps as part of a MCMC search is not viable, so a CUDA implementation of an ICIRAS method is detailed. The optimization strategies and safeguards developed to avoid erroneous magnifications being calculated are shown in detail, which result in an accurate and rapid, finite source magnification calculation.

Chapter 6

A web site alert and modelling system

6.1 Web site and modelling system overview

To visualize all the information that is created from the modelling of every microlensing event, a database and web site have been developed (http://www2.phys.canterbury.ac.nz/~u-lenser/). The web site is fully automated, showing the latest information in the database which comes from the ongoing modelling (Figure 6.1). The system autonomously retrieves the most recent data files from both MOA and OGLE survey telescopes, and any follow up observations made public. It also automatically links any events from each survey which are identified to be the same.

6.2 Autonomous single lens modelling system

Any new microlensing events which are added to the database are modeled as a single lens, following the methodology detailed in section 5.3. From the raw data, initial guess values need to be chosen for the model parameters. These initial guess values should be close to the true solution to aid in the convergence of the MCMC. If the initial guess values are too far from the global minimum, it frequently allows the MCMC chains to diverge as they try to explore a wider parameter space. This results in the chains taking a much longer time to converge to the global minimum.

To aid the automated system in starting its modelling in the correct area of parameter space, the current state of the microlensing event is used to determine the t_0 and t_E initial guess parameters. In the case of a rising event, where the latest data point has

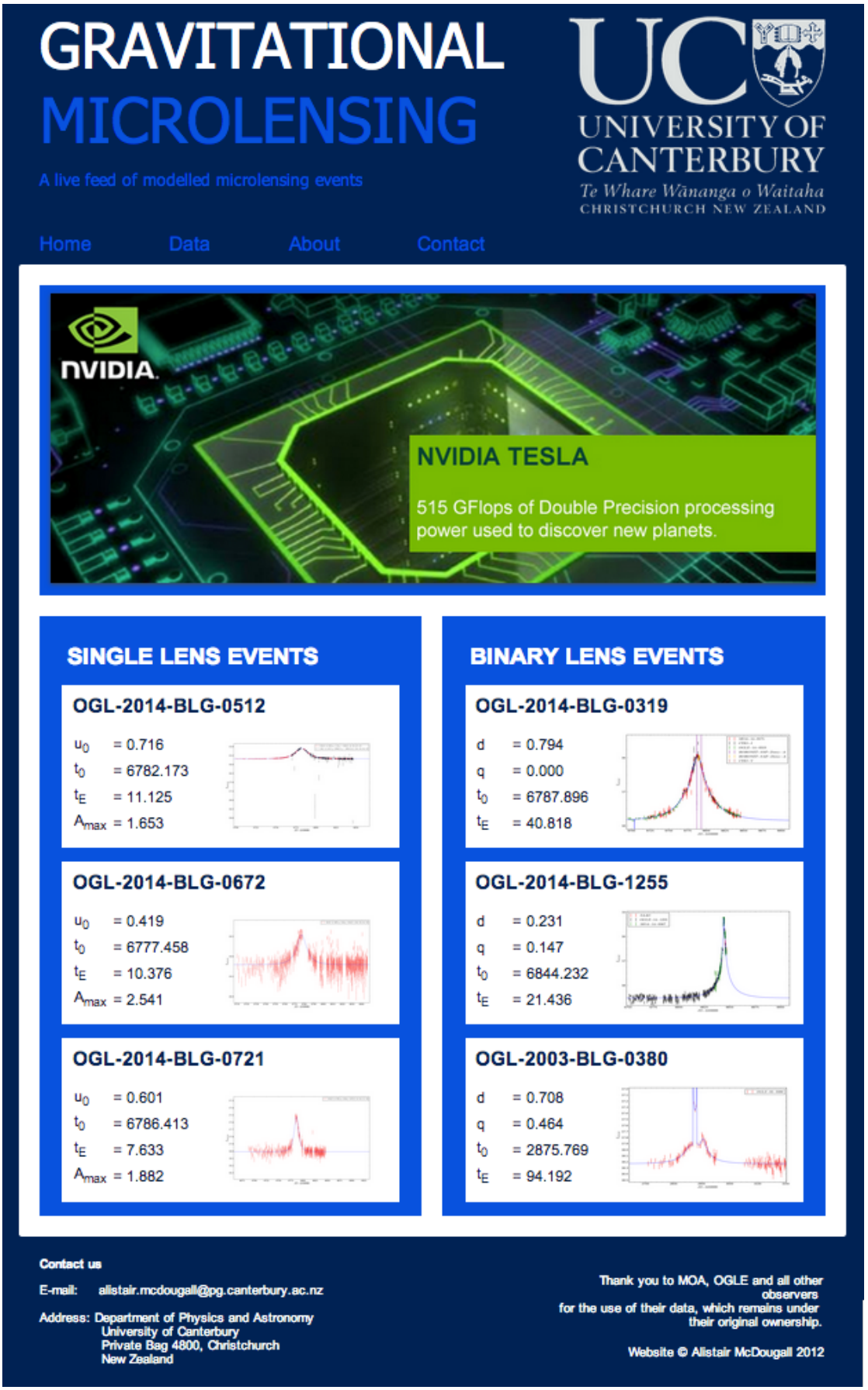


FIGURE 6.1: The home page of the microlensing web site, that provides access to the detailed models of each event. A list of the three most recently modelled single and binary lens events are displayed, which update with the latest results.

not yet crossed the data source's (either OGLE and MOA's web page alert system) predicted peak, t_0 is chosen to be 10 days ahead of the latest data point and t_E is set to be 22, which is around the average Einstein crossing time for a single lens event (Sumi et al., 2011). For an event which has passed the peak, t_0 is taken as the modeled value from the data source[1][2] and t_E is estimated from the data using a full width half maximum calculation. This is determined by starting from the data point closest to the peak and counting outwards in one direction, taking the average of four points. When the average of these data points fall below half the peak value, the time difference between the closest averaged point and the peak is taken as half the width. Repeating this process moving away from the peak in the opposite direction provides an estimate for the other half width, from which the full width can be determined by summing these two values together. If not enough data exists either on the rise or the fall of the event, t_E is estimated to be twice the value calculated for the other side's half width. In the more extreme case when neither side has enough data to determine the half width, an estimate of 22 days is made for the same reasons as a rising event. When calculating the half width, it is important to take the average of a few data points as opposed to a single value, because of noisy data.

In addition to modelling new events, each time an event is updated with new data, the system will automatically start remodelling it, however, due to limited computing resources the events are prioritized. A priority value is determined by the following algorithm,

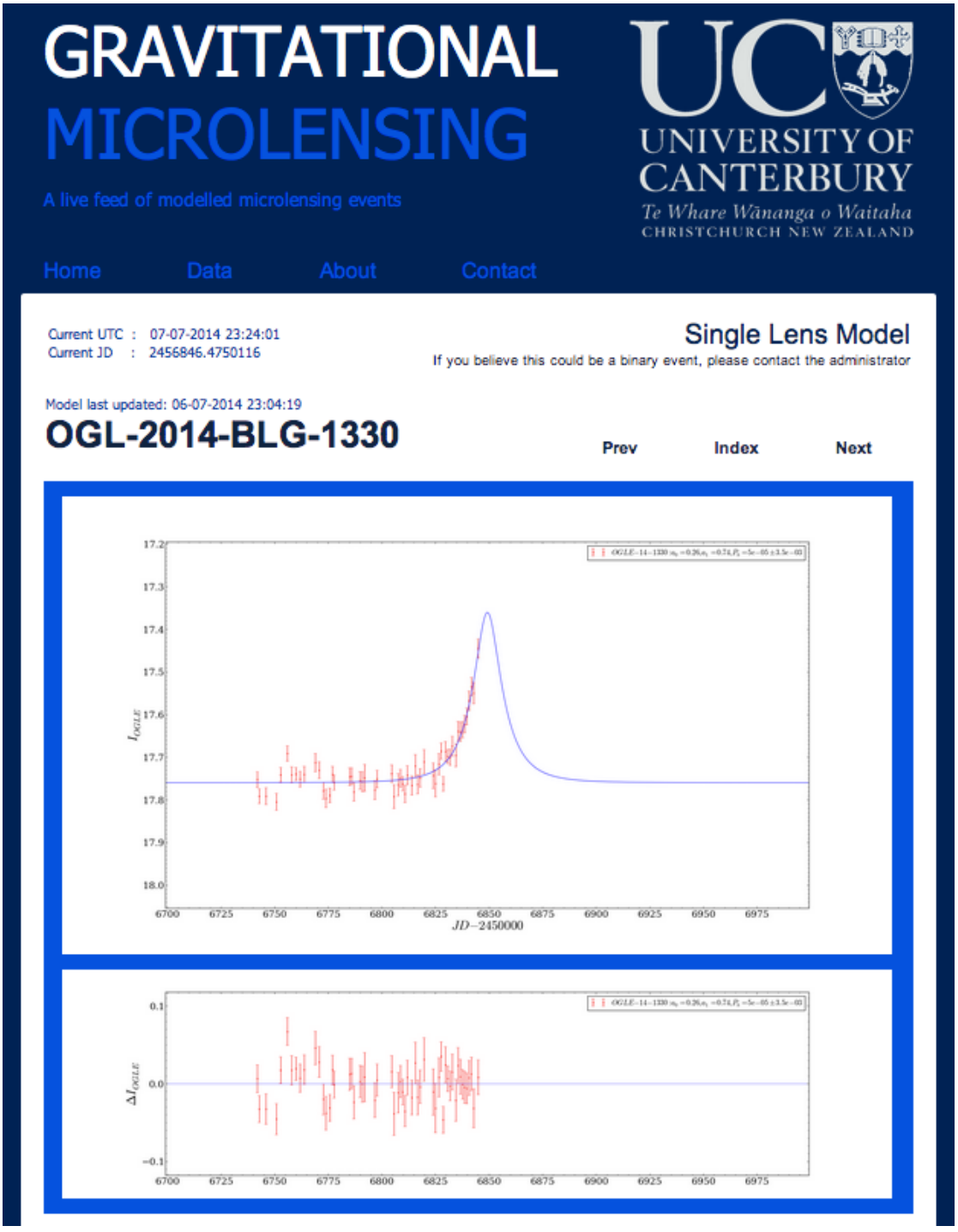
$$Priority = a + b + \frac{c}{|t_{now} - t_{model}|} \quad , \tag{6.1}$$

[1] All MOA data has been acquired from: https://it019909.massey.ac.nz/moa/alert/index.html
[2] All OGLE data has been acquired from: http://ogle.astrouw.edu.pl/ogle4/ews/ews.html

$$
\begin{aligned}
a &= \begin{cases} 1 & \text{new data} \\ 0 & \text{otherwise} \end{cases} \\
b &= \begin{cases} b_1 & \text{previous valid model} \\ b_2 & \text{otherwise} \end{cases} \\
b1 &= \begin{cases} \frac{10}{t_{E_{model}}} & \text{if } t_{E_{model}} \text{ is valid} \\ 10 & \text{otherwise} \end{cases} \\
b_2 &= \begin{cases} \frac{10}{t_{E_{estimate}}} & \text{if } t_{E_{estimate}} \text{ is valid} \\ 0.5 & \text{otherwise} \end{cases} \\
c &= \begin{cases} 1 & t_{now} > t_{model} \\ 0.5 & \text{otherwise} \end{cases}
\end{aligned}
\tag{6.2}
$$

where the first term only contributes to the priority if the event contains new data since it was last modelled. If the event has a valid t_E solution from previous modelling, it is used to contribute to the prioirty value, shown in the second term. In situations where no previous model exists, t_E estimates or fixed values are used. This contribution to the priority favours short time scale events, as they will complete quickly and therefore need to be modelled first. The final term adds to the priority based on how close to the predicted peak the current time is, with events that are still rising being favoured over those that have already passed the peak.

All single lens modelling is performed on the University of Canterbury, Department of Physics and Astronomy's Rocks cluster, which contains 7 intel i7 compute nodes, each with 8 3.40GHz processing cores. The headnode of this cluster has a python control program that identifies the highest priority event to be modeled by a single lens, and submits a job request to model it. The controller ensures no job is submitted multiple times and that there is minimal idle time on the cluster, by submitting new jobs based on the latest priority information the instant a job finishes. Upon completion of each modelling process the results are uploaded to the database and appropriate graphs are produced, showing the light curve, residuals and covariance from the MCMC processes (Figure 6.2). These results are instantly available through the web site, showing any user the most up to date model information for every microlensing event of the year. Due to the large number of events identified each year by the two survey telescopes, a single workstation is insufficient to produce all the required modelling, making the use of a cluster crucial.


GRAVITATIONAL
MICROLENSING
A live feed of modelled microlensing events
UC
UNIVERSITY OF
CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND
Home
Data
About
Contact
Current UTC : 07-07-2014 23:24:01
Current JD : 2456846.4750116
Single Lens Model
If you believe this could be a binary event, please contact the administrator
Model last updated: 06-07-2014 23:04:19
OGL-2014-BLG-1330
Prev
Index
Next
I_{OGLE}
$JD-2450000$
ΔI_{OGLE}

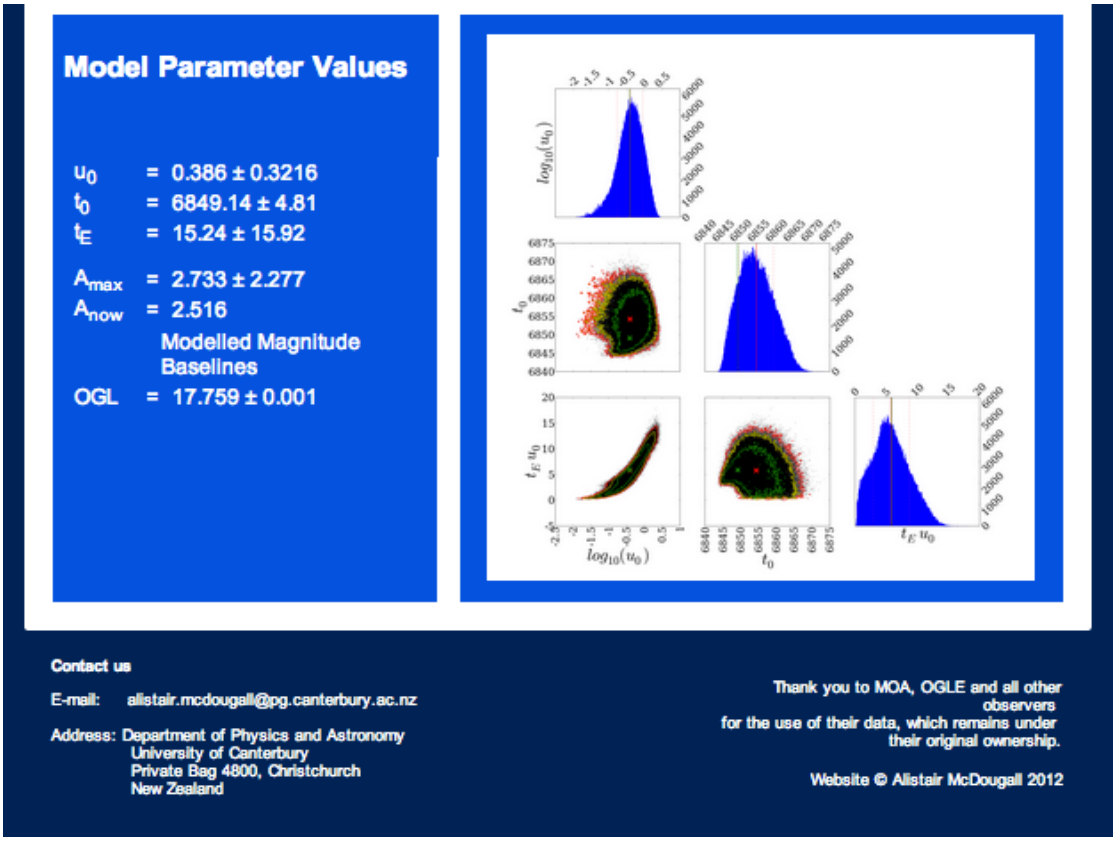


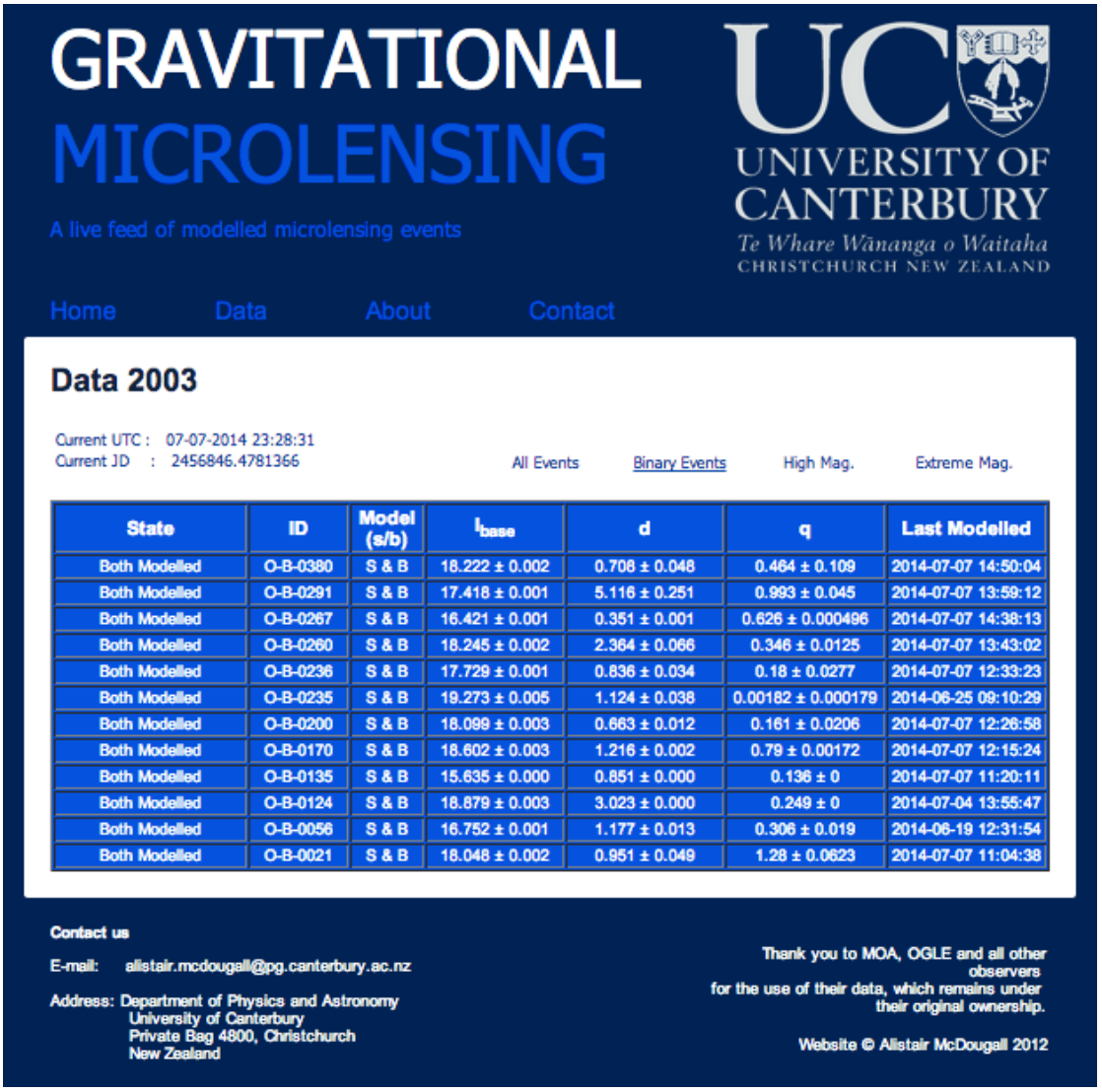
FIGURE 6.2: An example web page of a single lens event, showing the most likely model parameter set and the light curve and residuals it produces, as well as the covariance plots from the MCMC search.

6.3 Web site and database

The web site and database not only provide a user with the latest information about new microlensing events, but it also maintains an archive of all past events modeled, and automatically creates new pages each year to accommodate the new observing seasons. All recorded microlensing events can be shown as a list grouped by their calendar year. The list displays the event name, its baseline magnitude, modeled t_0, t_E, maximum magnification (A_{max}) and current magnification (A_{now}), as well as if the event is currently rising or falling. If an event has sufficiently small errors and peak magnification large enough to exceed a threshold ($A_{max} > 10$), then it is flagged as an important event and is highlighted in the list. Similarly if the maximum magnification exceeds fifty with an appropriately small error it is flagged as very important and is again identified clearly in the list. Any event from the list can be selected, which leads to a detailed page of results for that event (Figure 6.2). The page shows the light curve, residual, and covariance plots alongside the full parameter set and associated errors, it also identifies if the event is included in the list under a different name, which occurs when both of the survey telescopes observe the same microlensing event.

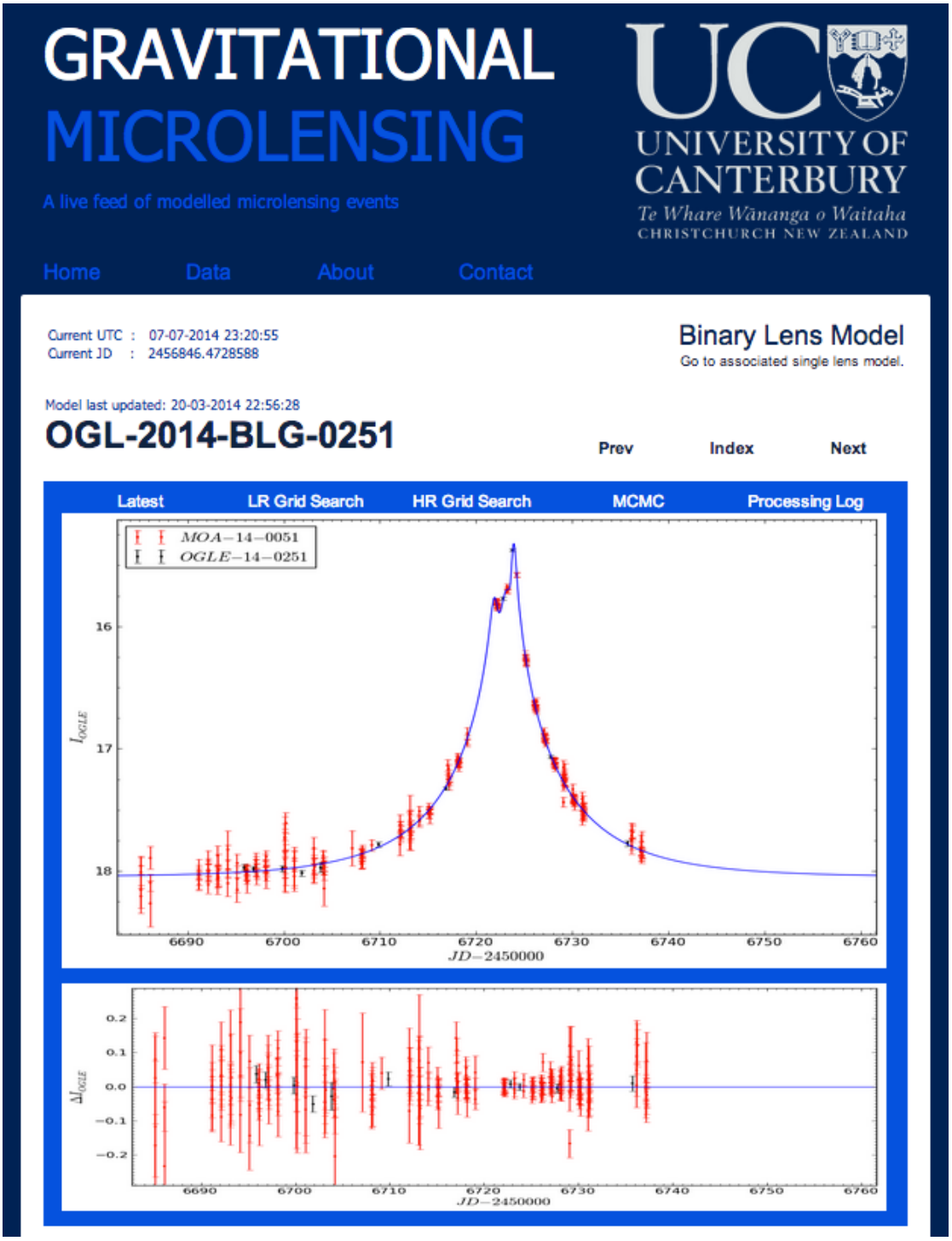
Single lenses cause the majority of microlensing events, but there also exist those that are caused by binary or larger lens system. If a user suspects an event to be caused by a binary lens, it can be identified and modeled, which proceeds to determine the most likely binary lens solution. Through the web site it is possible to list all identified binary lens events of each year (Figure 6.3). This displays each event's baseline magnitude, the model's d, q parameters, and the date and time when it was last modeled. By selecting a binary lens event it is possible to access the detailed output from its modelling process. The web page displays the latest model solution parameter set, its associated light curve, residuals, caustic map with the source trajectory overlaid, and an additional figure detailing the latest stage of the modelling process. As the binary lens modelling process is made up of several stages (detailed earlier in Chapter 5.6), the results of each stage can be viewed in separate windows (Figure 6.4). Each window displays the stage's best parameter set, light curve, residuals, caustic map with overlaid source trajectory and additional figures dependent on the stage of the modelling process. For the initial d, q grid search (section 5.6.1), a minimized d, q map is displayed showing areas of local minima. The results from this stage initializes the second part of the modelling process, the local MCMC chains (section 5.6.2). The second display window has the same format as the first, but with the d, q grid being overlaid by the MCMC chain movements, highlighting which chain results in the best solution. The final stage of the modelling process (section 5.6.3.10) is only initialised upon user request, where it performs a thorough MCMC search around the minimum found by the local area

MCMC search, it also calculates the standard deviation of each parameter in the solution. The resulting window of information has the same form as previously, but instead of a minimized d, q map, it displays a covariance matrix of all parameters from the movement in the MCMC process. A final window of information is provided for each binary lens event that displays the processing log file, which is updated during the modelling process. This file contains information about what the modeler did, the results at each stage, and all timing information. The processing log is continually updated during a modelling process, displaying the current percentage complete, time taken, as well as an estimate for the time to completion of each stage. Throughout the modelling process each window will update with the latest results, keeping a user up to date with the current progress of the modelling process.



State	ID	Model (s/b)	I_{base}	d	q	Last Modelled
Both Modelled	O-B-0380	S & B	18.222 ± 0.002	0.708 ± 0.048	0.464 ± 0.109	2014-07-07 14:50:04
Both Modelled	O-B-0291	S & B	17.418 ± 0.001	5.116 ± 0.251	0.993 ± 0.045	2014-07-07 13:59:12
Both Modelled	O-B-0267	S & B	16.421 ± 0.001	0.351 ± 0.001	0.626 ± 0.000496	2014-07-07 14:38:13
Both Modelled	O-B-0260	S & B	18.245 ± 0.002	2.364 ± 0.066	0.346 ± 0.0125	2014-07-07 13:43:02
Both Modelled	O-B-0236	S & B	17.729 ± 0.001	0.836 ± 0.034	0.18 ± 0.0277	2014-07-07 12:33:23
Both Modelled	O-B-0235	S & B	19.273 ± 0.005	1.124 ± 0.038	0.00182 ± 0.000179	2014-06-25 09:10:29
Both Modelled	O-B-0200	S & B	18.099 ± 0.003	0.663 ± 0.012	0.161 ± 0.0206	2014-07-07 12:26:58
Both Modelled	O-B-0170	S & B	18.602 ± 0.003	1.216 ± 0.002	0.79 ± 0.00172	2014-07-07 12:15:24
Both Modelled	O-B-0135	S & B	15.635 ± 0.000	0.851 ± 0.000	0.136 ± 0	2014-07-07 11:20:11
Both Modelled	O-B-0124	S & B	18.879 ± 0.003	3.023 ± 0.000	0.249 ± 0	2014-07-04 13:55:47
Both Modelled	O-B-0056	S & B	16.752 ± 0.001	1.177 ± 0.013	0.306 ± 0.019	2014-06-19 12:31:54
Both Modelled	O-B-0021	S & B	18.048 ± 0.002	0.951 ± 0.049	1.28 ± 0.0623	2014-07-07 11:04:38

FIGURE 6.3: An example web page of a year of recorded binary lens events.


GRAVITATIONAL
MICROLENSING
A live feed of modelled microlensing events
UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND
Home
Data
About
Contact
Current UTC : 07-07-2014 23:20:55
Current JD : 2456846.4728588
Binary Lens Model
Go to associated single lens model.
Model last updated: 20-03-2014 22:56:28
OGL-2014-BLG-0251
Prev
Index
Next
Latest
LR Grid Search
HR Grid Search
MCMC
Processing Log
MOA−14−0051
OGLE−14−0251
16
17
18
I_{OGLE}
6690
6700
6710
6720
6730
6740
6750
6760
JD−2450000
0.2
0.1
0.0
−0.1
−0.2
ΔI_{OGLE}
JD−2450000

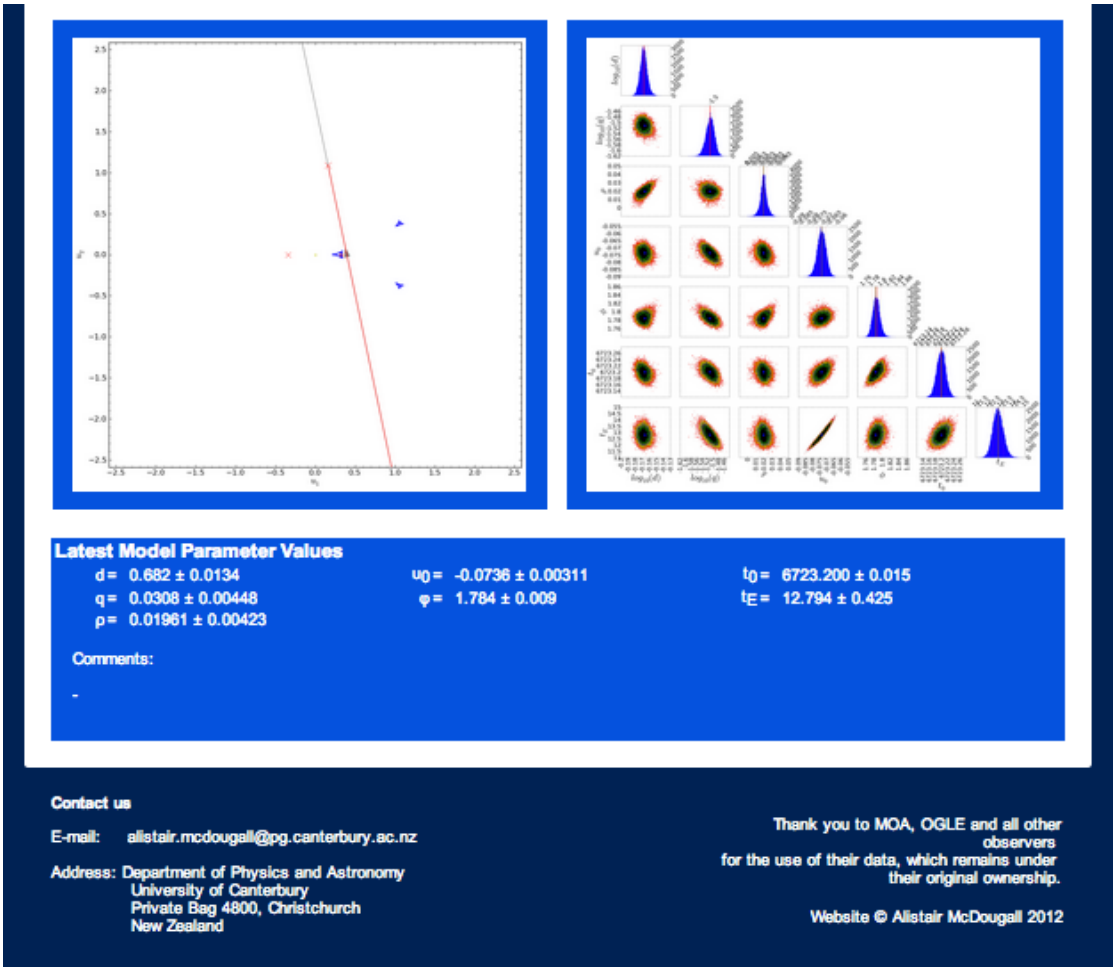


FIGURE 6.4: An example web page of a binary lens event, showing the most likely model parameter set and the caustic with overlaid source trajectory, light curve, and residuals it produces, as well as the covariance plots from the MCMC search.

6.4 Modelling system control

The web site provides a user the ability to monitor all current microlensing events, by displaying the live analysis of both single and binary lens events. However, the web site is only a front end used to view the information of the modelling processes, the data management happens in the background. For user interaction with the modelling system a control software has been developed to run on most Unix based machines (Figure 6.5). It provides a user with additional information and full management of the single and binary lens modellers. The control software is divided into sections; single lens events, binary lens events, and database management.

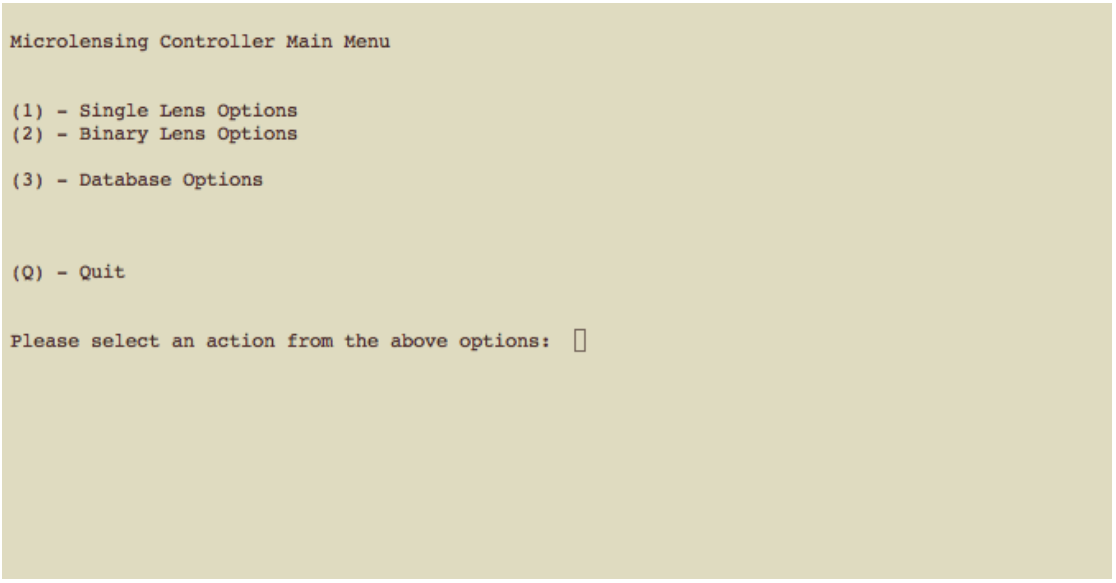


FIGURE 6.5: The main page of the microlens modelling system's control software, displaying the possible options to a user.

6.4.1 Single Lens Controller

The single lens control set (Figure 6.6) allows the user to list all events currently queued and those being modeled on the Rocks Cluster. Should the user want to model a specific event, they are able to submit a new job request to the cluster, as long as the event is not already being modeled or currently in the queue. As the single lens modelling is achieved using MCMC methods, it is possible to provide an initial t_0 and t_E estimate which are used to distribute the starting parameter values around. An additional option exists that allows the user to provide an e-mail address should they wish to receive a confirmation e-mail upon completion of their requested job. The e-mail contains a link to the web page displaying the latest results and figures, and attaches the output files created by the processed job.

Every completed job automatically produces plots of the results, but these figures are not always as wide or as focused as the user wants. Therefore an option exists for the user to re-plot the figures of any single lens event using the current model parameters stored in the database. Upon requesting new plots, it is possible to provide details about how many data points to include and the dimensions of the figure's axis. The requested figures are then rapidly produced and updated so that they can be viewed through the web site. This tool is particularly useful for the close inspection of the model's peak, which is important when looking for small deviations from the model that may be caused by a binary lens.

Further control of the single lens modelling provides the ability to stop the modelling of an individual single lens event or all single lens events. Additionally, if a result is known to be wrong, or an error occurred, the controller software enables a user to remove a model from the database and clear the web page of invalid figures.

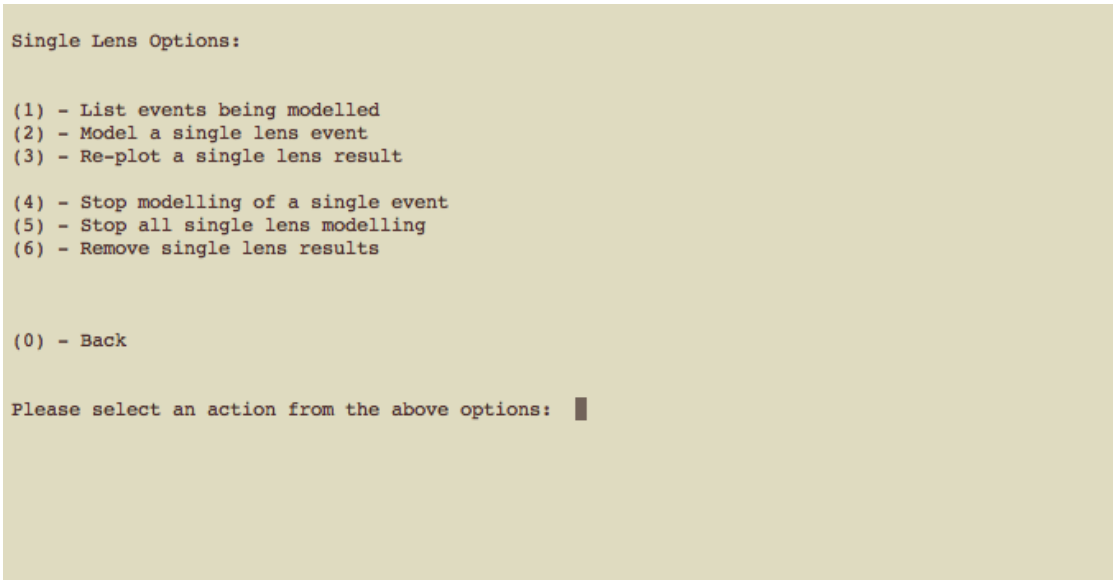


FIGURE 6.6: The single lens events main page of the microlens modelling system's control software, which provides a variety of modelling and job management controls.

6.4.2 Binary Lens Controller

The least automated part of the modelling system is the recognition and initialization of binary lens events, therefore the largest number of user controls can be found within the binary lens controller (Figure 6.7). By default all microlensing events are assumed to be a single lens, however, if a user recognizes deviations from a single lens model they can identify the event as a possible binary lens, which flags it in the database, causing it to appear in the binary lens list, and allowing binary lens modelling and further analysis. Similarly should an event be incorrectly marked as binary it is possible to remove the identification, returning it to a single lens event and removing the web page detailing any binary lens analysis. Should something anomalous happen or there is additional information about an event that needs to be retained; a comment section exists which is displayed on the binary lens events web page. A user is able to view and modify this content which will automatically update on the web site.

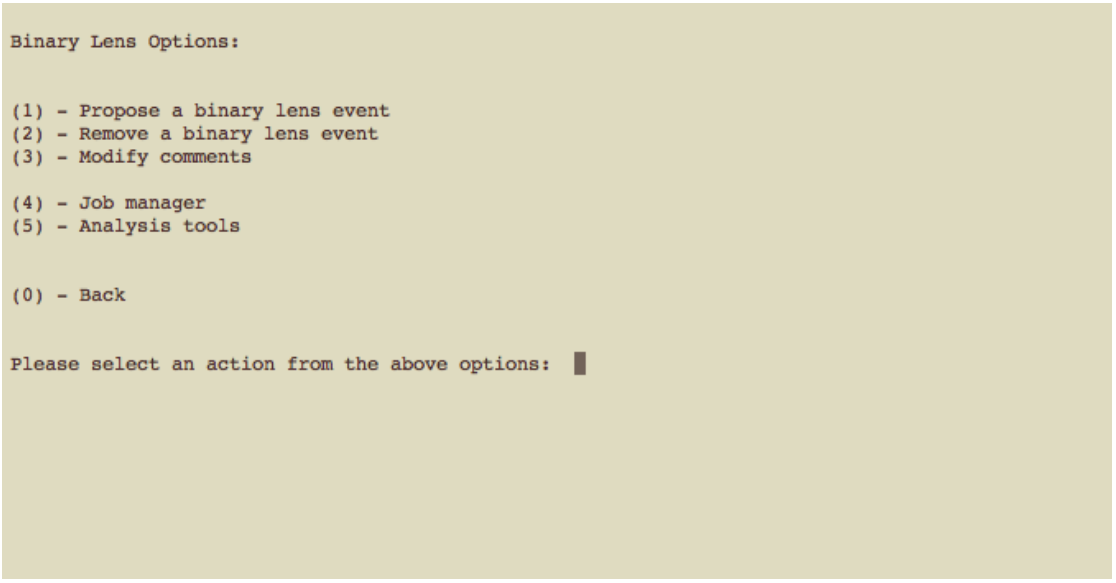


FIGURE 6.7: The binary lens events main page of the microlens modelling system's control software, which provides options to identify new or remove binary lens events and access more job management or analysis tools.

6.4.2.1 Job Manager

Once an event is identified as a binary lens, the control software is able to initialize the binary lens modelling process (Figure 6.8); this requires the input of a minimal amount of information before it performs a thorough analysis of parameter space. The user input identifies if the data follows the general shape of a Paczyński curve without any major anomalies on a shoulder, often caused by a source passing close to a central caustic making the Paczyński curve asymmetric. If the data follows this trend then the user is required to input an estimate for the t_0 and t_E parameters (which can come from its single lens model solution). Alternatively if the data clearly deviates from the Paczyński curve, the user is required to input how many peaks they can see in the data, or know will exist in the model if the event is still ongoing. They are then required to provide the modified Julian Date ($JD - 2450000$) of the most defined peak(s) in the data, a minimum of one peak is required but up to two can be provided to further constrain the possible models and reduce computation time. Additional information that can be provided to enhance the modelling and reduce the search time is the minimum limit for the maximum magnification, achieved by providing a conservative measurement of the largest magnitude difference in the data. An option to exclude individual data sets exists, allowing the user to model an event based only on the data sources of their choosing. To ensure the modelling encompasses all the data the user wants, a specific Julian Date range can be provided. The inclusion of excess data that does not aid in constraining the model causes additional calculations which reduce the performance of

the modelling process, therefore a maximum number of data points per source needs to be provided. Due to hardware constraints, an upper limit of 7168 data points per data source exists. Should more than this be requested the earliest dated points will be truncated to ensure the limit is not exceeded. Occasionally a full analysis of an event is not required, only an initial search of parameter space to identify areas of local minima. An option therefore exists to exclude the final EMCEE part of the search, the process which fine tunes the search solution and provides error estimates but is time consuming. As with the single lens modelling a final option exists for the user to provide an e-mail address to receive a confirmation e-mail upon completion of the job, with the complete processing log file attached and a link straight to the binary lens event's result page.

All binary lens modelling is performed on a custom built GPU workstation which has a limited number of GPUs, resulting in a limit to the number of jobs that can be processed simultaneously. Therefore a queue management system has been developed. Should too many jobs be submitted they are added to the queue and processed as soon as the computing resource becomes available.

To see the current state of all binary lens modelling, it is possible to retrieve a list of all binary lens events which are being modeled on which GPU device, the list also display the full queue of binary lens events awaiting a free GPU device. For detailed information about the currently installed GPU devices, an option exists to return a breakdown of each device including its power consumption, temperature, memory usage and more. It also shows how many GPU devices remain free for binary lens modelling.

Basic control of this queue management exists, allowing a user to view, remove and re-order the events currently waiting to be modeled.

6.4.2.2 Analysis tools

Upon completion figures are made to show the result of the modelling process, however, sometimes this does not show an area of the light curve in enough detail. Therefore as with single lens models, the user is able to re-produce the plots for a binary lens event inputing details about what to show. Using the model parameters determined at each stage of the modelling process and the latest data available, the user can provide limits of the axis, data sets to exclude, a date range of the data to include, and an upper limit to the number of data points. The updated plots of a binary lens event can be processed at any time, even during a current modelling process. This produces all the same figures using the most recent best solution parameter set currently found in the ongoing modelling process and displays the current progress in the d, q grid search map.

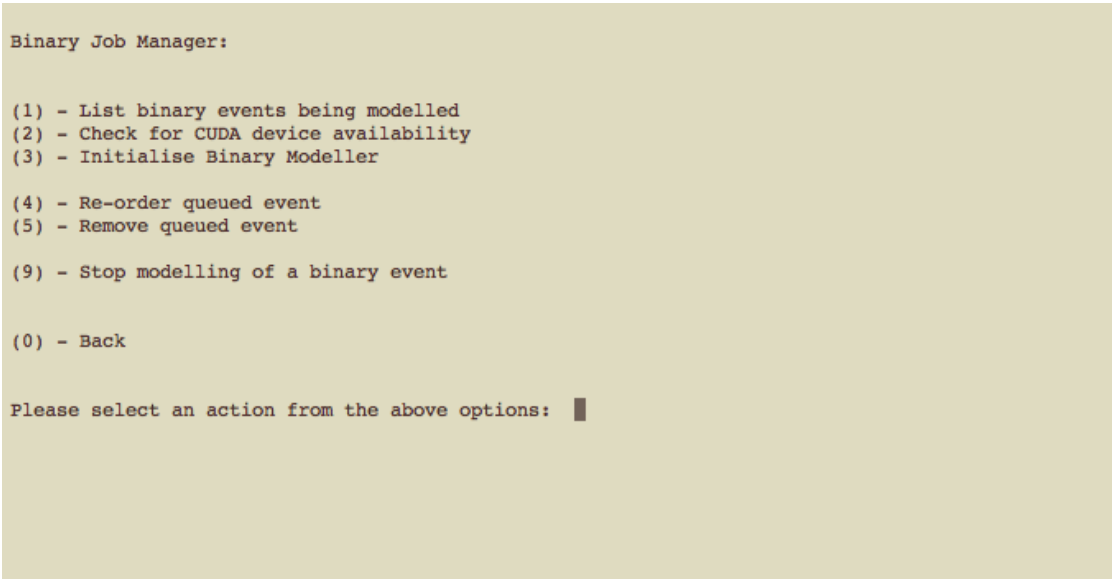


FIGURE 6.8: The binary lens events job manager page of the microlens modelling system's control software, which provides options to list, re-order, add, and remove events currently being modelled or in the binary lens modelling queue.

After the modelling of an event is complete, further analysis is often required to check the validity of the model. Such analysis tools are provided through the controller (Figure 6.9) enabling the user to perform a comparison between the single lens model and the binary lens model of an event, and a comparison between the search result's best parameter set and a user defined parameter set. Both these analysis tools have the same required input information options as for re-plotting, with the addition of a user defined binary lens parameter set. The output of this tool presents side by side comparison figures showing the binary lens search solution light curve, and residuals next to the desired comparison model and its residuals. In the case of the user defined binary lens model comparison, an additional figure showing the different caustic structures and source trajectories overlaid on top of each other is also produced.

6.4.3 Database management

The final section of the system controller allows the user to view and manage the database of all stored microlensing events (Figure 6.10). The ability to view every piece of information in the database for a single event, add, remove and modify any event, enables full control of the database that drives the web site.

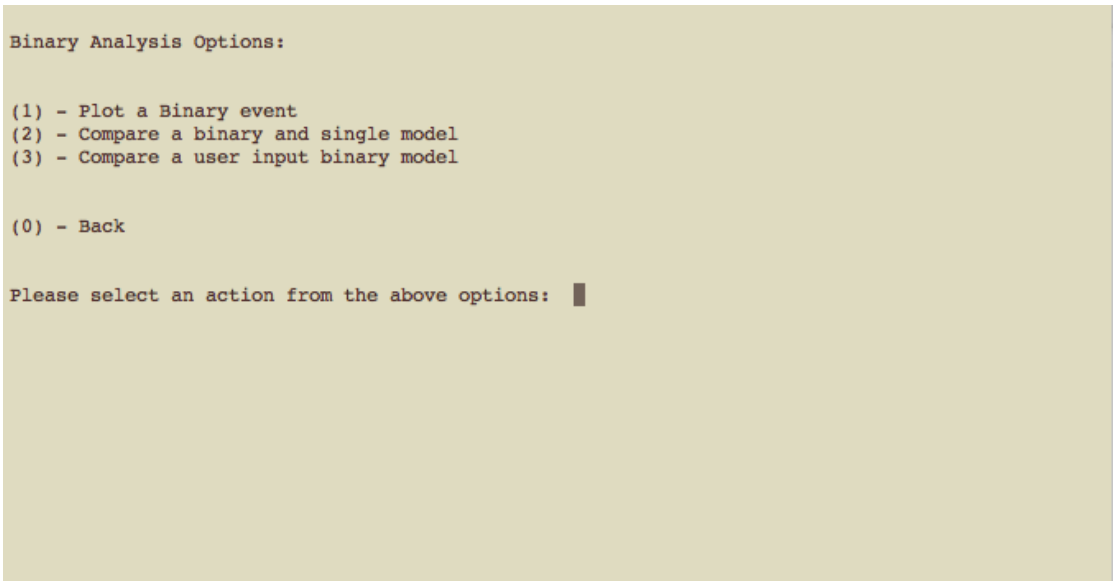


FIGURE 6.9: The binary lens events analysis tools page of the microlens modelling system's control software, which provides options to re-plot, or perform model comparisons.

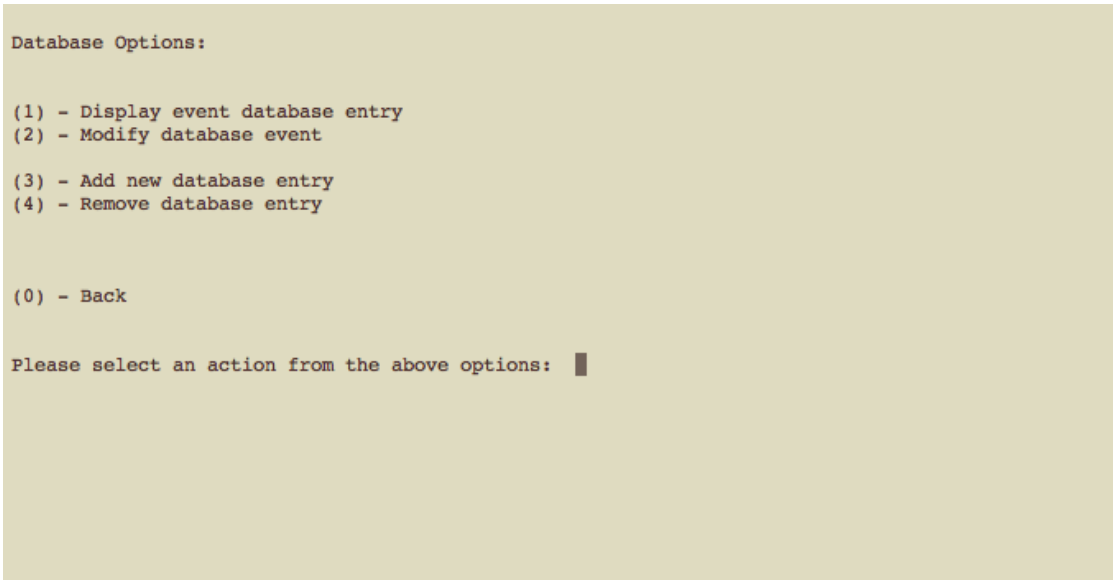


FIGURE 6.10: The database control main page of the microlens modelling system's control software, which provides options to view, modify, add, and remove events in the database.

6.5 Summary

As a means to visualize and monitor all the microlensing events being modeled, a web site has been developed to publish, and archive all events modelled to date[3]. A detailed description of the web pages available and what information they contain is provided in this chapter. The web site is presented in such a way that any user can identify the latest single and binary lens models produced, which can be used to help guide decisions about follow up observations.

As a means of starting the modelling of any event or analyzing the models produced, a control software has been developed. This is also presented in this chapter, detailing what features are available and how it works. A suite of control tools exist for both single and binary lens events that enable the initializing of each modelling procedure, re-plotting of multiple figures, and analysis tools such as a single lens binary lens comparison. In addition to these modelling tools are some more administrative controls, such as inspecting and re-ordering the modelling queues, and the ability to view and modify the MySQL database that sits behind the web site, recording all the model outputs and event priorities.

[3] http://www2.phys.canterbury.ac.nz/~u-lenser/

Chapter 7

Model validation — Past binary lens results

7.1 Overview

This chapter details the analysis of several binary lens microlensing events. Using the techniques detailed in Section 5.6, a search of the seven dimensional parameter space is made to determine the best fitting solution to the observed data. The events analysed (OGLE-2012-BLG-0406, MOA-2003-BLG-0053/OGLE-2003-BLG-0235, and 14 events from the 2002-2003 OGLE seasons) have previously been studied and are referenced so that a comparison and confirmation of the modelling method presented in this thesis can be made.

Unless otherwise stated, all analysis is performed on the data made publicly available through each of the microlensing groups' web sites, and a linear limb-darkening coefficient of $\Gamma_I = 0.53$ is used. An Altered Julian Date (AJD) is defined to be equal to JD - 2450000, which is used by the modeller and quoted in the following chapters. From this data, a single lens model is determined using the methods detailed in Section 5.3. Using the single lens model can help to determine an initial estimate for two of the binary lens parameters if required; the time of closest approach (t_0) and the Einstein radius crossing time (t_E). Providing a few input values to the binary lens modeler starts the grid search of the seven dimensional parameter space, which locates areas of probable solutions to be explored with localized MCMC.

7.2 Binary lenses in OGLE-III EWS database 2002-2003

In 2004 Jaroszyński et al. (2004) published a paper detailing 15 binary lens events from the 2002-2003 OGLE observing season, which followed on from his earlier statistical study (Jaroszyński, 2002) of microlensing events observed during the OGLE seasons 1997-1999. Taking the events detailed in his 2004 paper, we perform an analysis of each event using the data provided directly from the OGLE EWS system without any error scaling or re-reduction to simulate a situation where the anomaly has recently been discovered and no known solution exists. This analysis is performed to show a rapid modelling of each event, which is able to correctly identify the area of global minimum. A direct χ^2 comparison between the models of Jaroszyński et al. (2004) and those presented here cannot be made due to his solutions being with error re-scaled data over an unknown time range, and in most cases a point source model. M. Jaroszyński's solutions are fit using a single linear fit parameter to describe the blending at a baseline value which is not provided and the binary lens parameters are presented to a fixed number of significant figures with no error. Therefore, upon modelling his parameter solutions with a different set of data it is sometimes necessary to perform a minimization method to locate the minimum in the vicinity of the solution he presented. In situations where the observations cover a caustic crossing, the parameter solutions of M. Jaroszyński may fit very poorly due to the very deep and narrow nature of the parameter space causing the slightest change in the data (from rescalled errors) or parameter values (from the rounding of significant figures) to move the solution outside of the χ^2 valley. Despite this lack of information it is still possible to identify if the new solutions presented locate the same area of global minimum, and if not, an attempt to identify the difference is made.

Below we present 14 of the models detailed by Jaroszyński et al. (2004) showing the global minimum and time taken to locate it. One of the events (OGLE-2003-BLG-0235) has been excluded from this list as it has been analyzed in greater detail (Section 7.3).

7.2.1 OGLE-2002-BLG-0051

Modelling of this event was initialized by stating that the model is required to have at least two peaks located around the AJDs of 2384.82062 and 2424.74, and a minimum change in magnitude of at least 2.5. The model was provided with 112 data points between the AJDs of 2100.0 and 3000.0. A grid search and local MCMC exploration (Figure 7.1) took a total of 1 hour and 53 minutes to determine the location of the global minimum, which was followed by the EMCEE search method on error-normalized data

to determine the error terms, shown in Table 7.1, This parameter set produces the caustic structure and trajectory, light curve, and residuals of Figures 7.2-7.4.

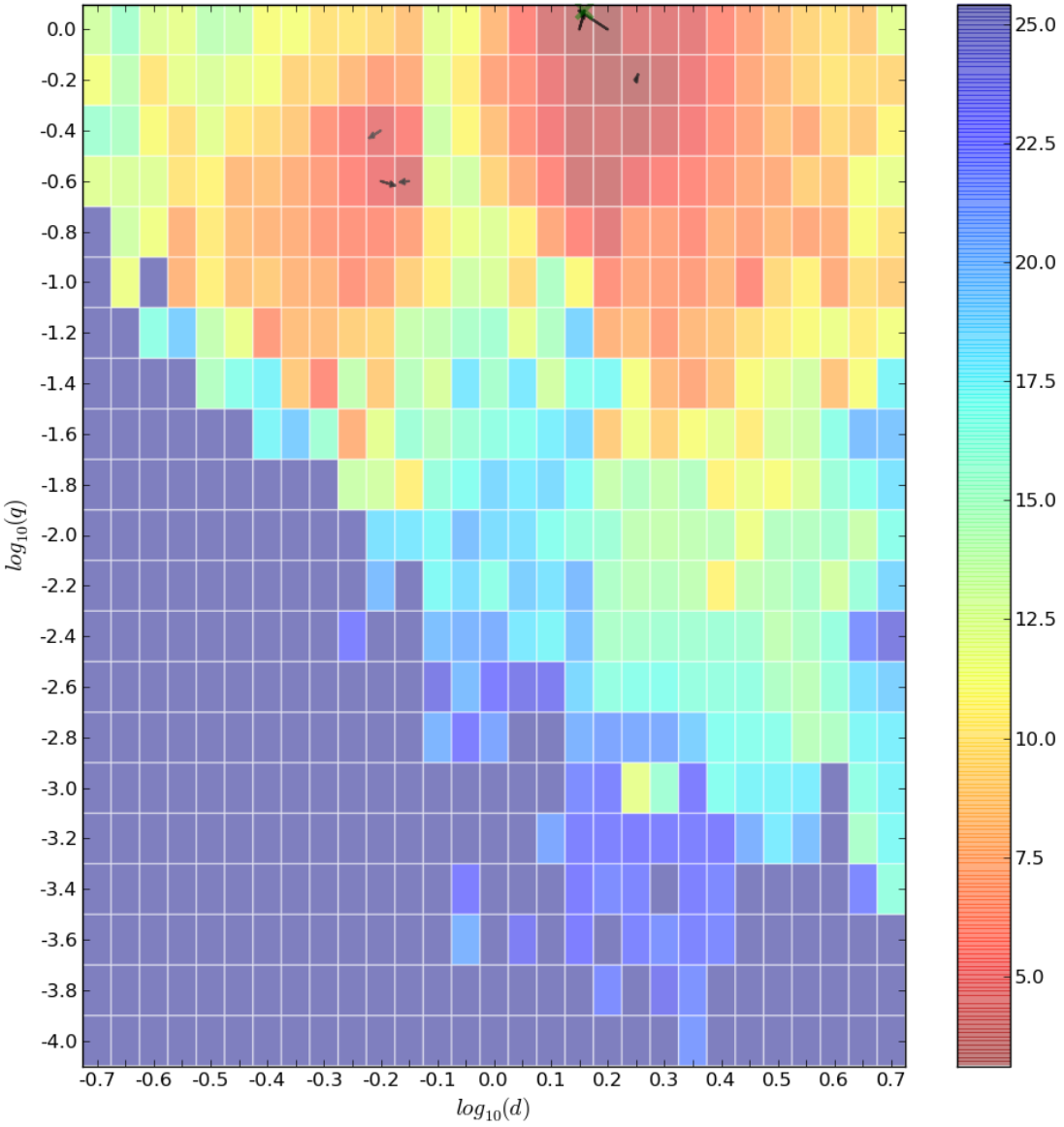


FIGURE 7.1: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2002-BLG-0051.

TABLE 7.1: Binary lens model solutions of OGLE-2002-BLG-0051, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	1.390	1.38993 ± 0.00070	1.44 ± 0.11
q	0.943	0.94384 ± 0.00097	1.18 ± 0.074
ρ	–	1.1×10^{-7}	0.0018 ± 0.00012
u_0	0.48	0.4775 ± 0.0015	-0.483 ± 0.037
ϕ	-1.6879	-1.68785 ± 0.00014	4.83 ± 0.11
t_0	2391.4	2391.424 ± 0.069	2392.2 ± 4.9
t_E	88.1	88.146 ± 0.095	96.1 ± 4.0
χ^2	782.215	298.744	291.663

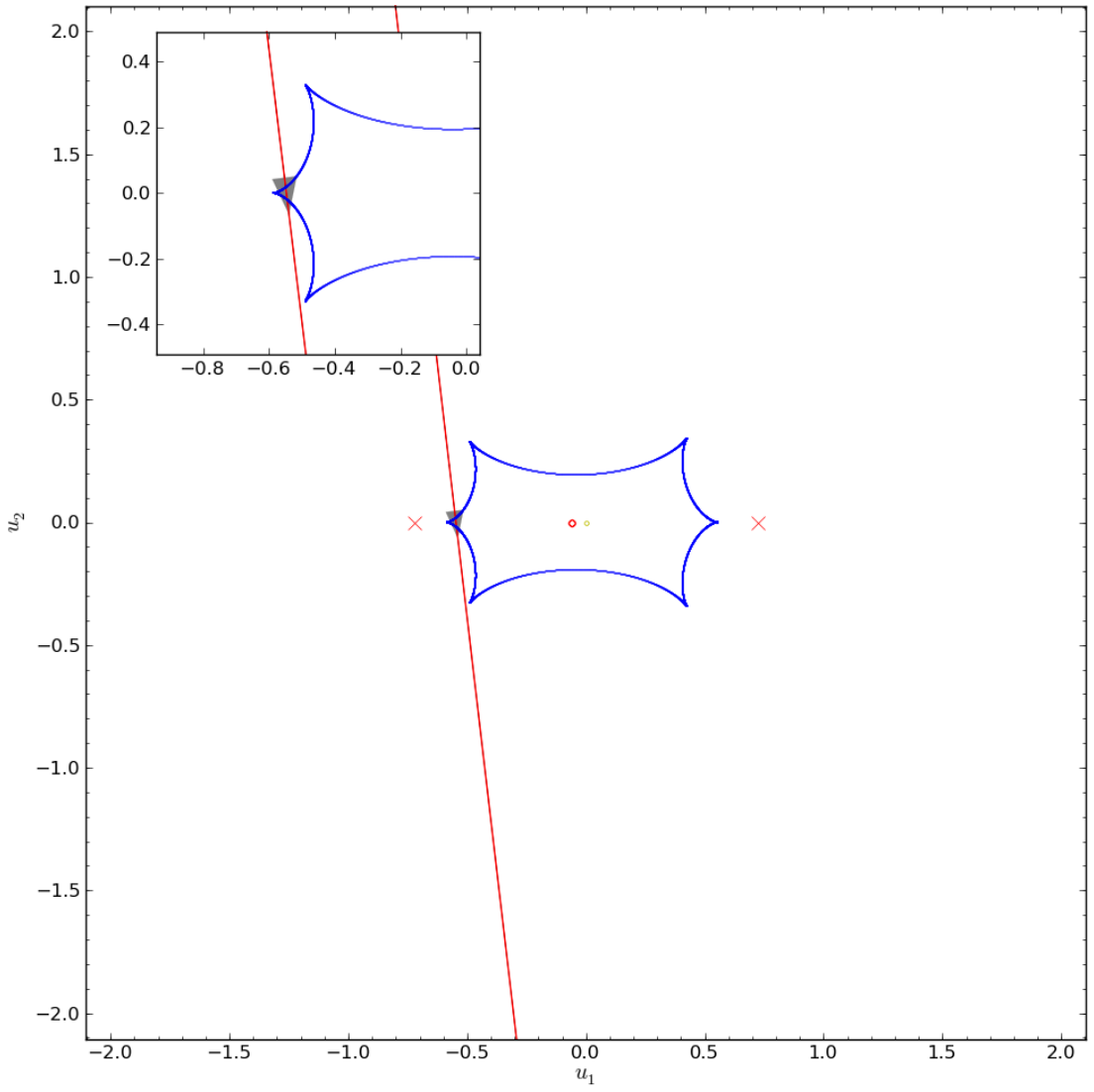


FIGURE 7.2: A caustic map with overlaid trajectory for the model solution (Table 7.1) of OGLE-2002-BLG-0051determined by the GPU search method.

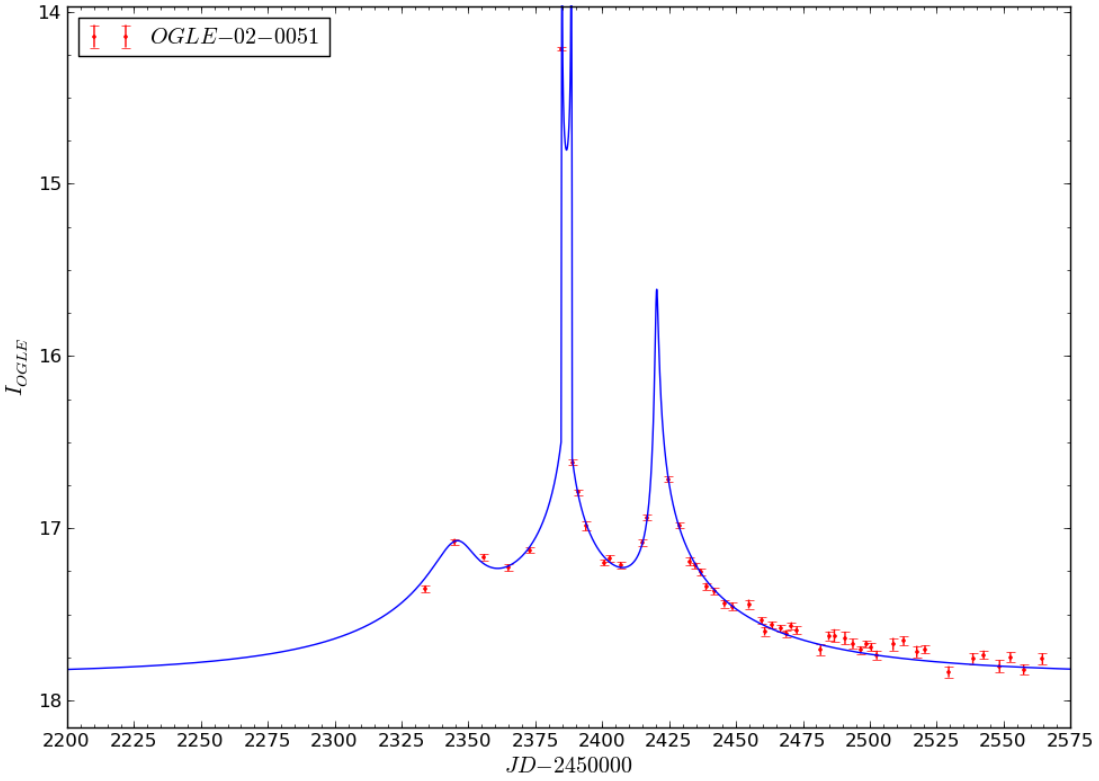


FIGURE 7.3: A light curve of the model solution (Table 7.1) of OGLE-2002-BLG-0051 determined by the GPU search method.

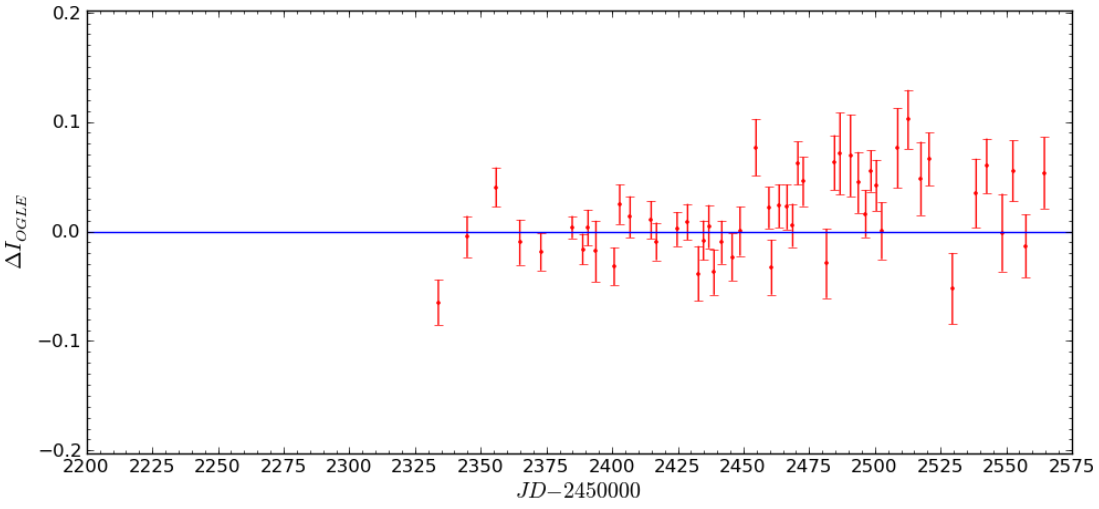


FIGURE 7.4: Light curve residuals of the model solution (Table 7.1) of OGLE-2002-BLG-0051 determined by the GPU search method.

The global minimum area is in agreement with the solution presented by M. Jaroszyński shown in Table 7.1. Although q and ϕ appear different, this comes from the choice of which lens mass is larger ($q < 1$ or $q > 1$). Taking the smaller q value (M. Jaroszyński's solution) as $1/q$, the caustic structure is mirrored in the y-axis, and therefore ϕ is modified appropriately ($\phi + \pi$), producing a comparable parameter set to the GPU presented solution.

7.2.2 OGLE-2002-BLG-0069

This event was modeled by providing a required minimium change in magnitude of at least 2.8, and for the model to contain at least two peaks, located near the AJDs of 2453.67882 and 2462.54548. The data used for the modelling process included 106 points between the AJDs of 2127.0-3055.0. A grid search and MCMC exploration of six local minima (Figure 7.5) took 2 hours and 13 minutes to complete before an error-normalized EMCEE search of the global minimum was performed to determine the error terms for each parameter, detailed in Table 7.2, which produce the model's caustic, light curve, and residuals Figures of 7.6-7.8.

TABLE 7.2: Binary lens model solutions of OGLE-2002-BLG-0069, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with those determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU close	GPU wide
d	0.497	0.4987 ± 0.0028	0.49471	3.891 ± 0.044
q	0.721	0.7149 ± 0.0021	0.75714	0.1679 ± 0.0031
ρ	–	1.1×10^{-7}	0.00458	0.002098 ± 0.000038
u_0	-0.02	-0.0195 ± 0.0059	-0.01371	2.769 ± 0.040
ϕ	1.927	1.92604 ± 0.00049	1.93271	1.093 ± 0.010
t_0	2456.4	2456.25 ± 0.30	2456.59	2898 ± 13
t_E	99.7	99.21 ± 0.26	99.78	308.7 ± 8.7
χ^2	1050.893	1018.243	134.318	126.329

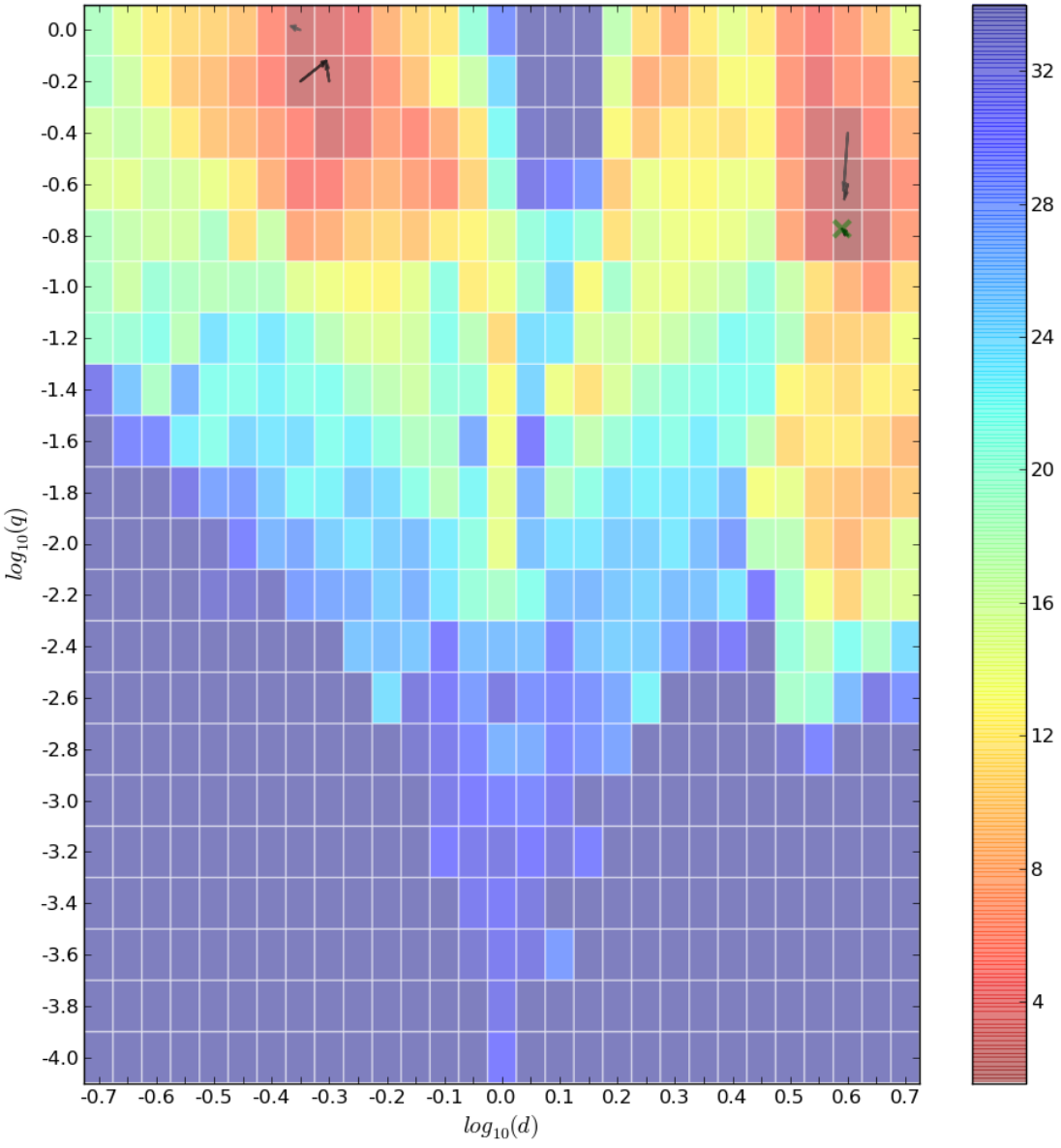


FIGURE 7.5: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2002-BLG-0069.

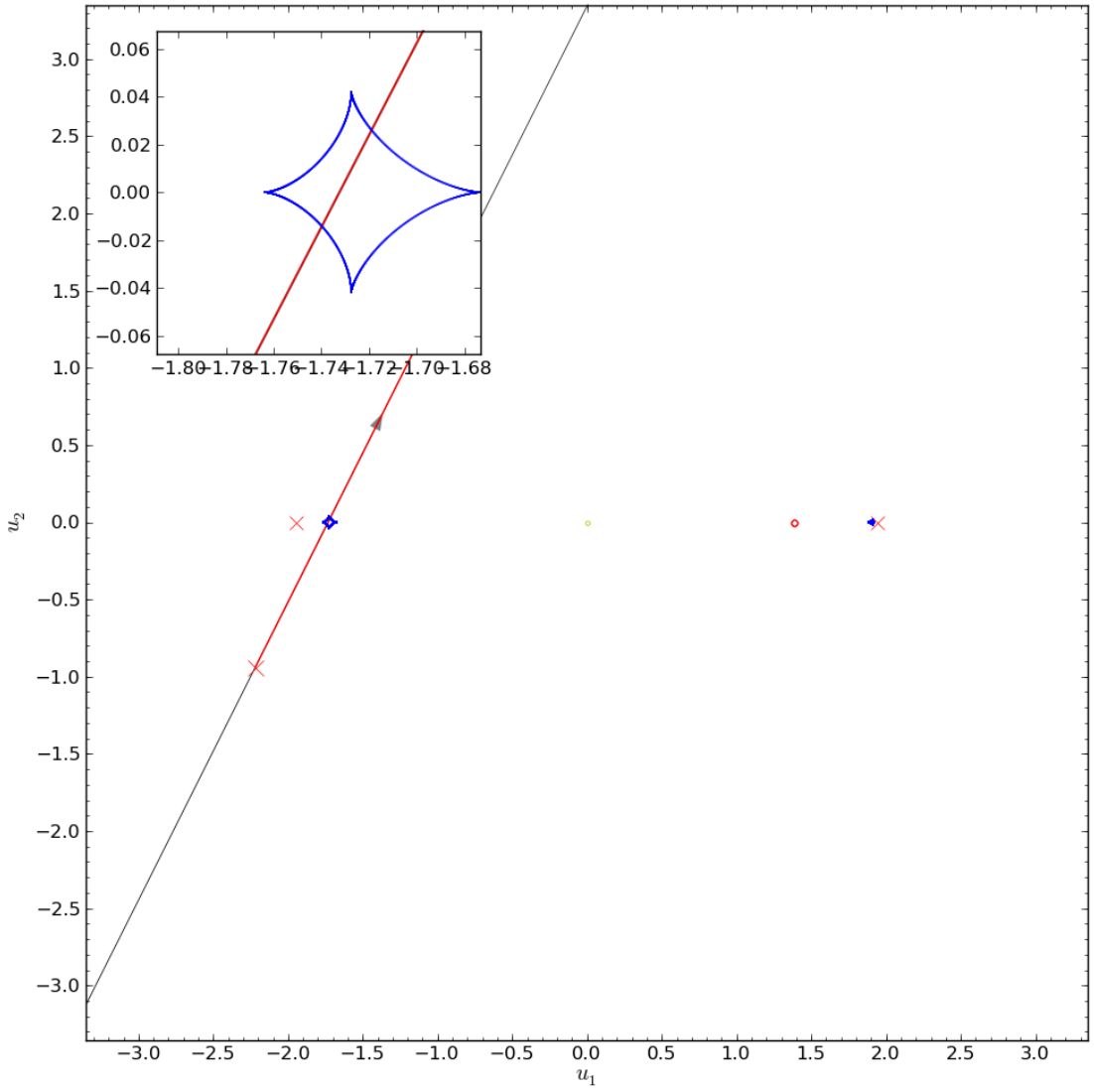


FIGURE 7.6: A caustic map with overlaid trajectory for the model solution (Table 7.2) of OGLE-2002-BLG-0069 determined by the GPU search method

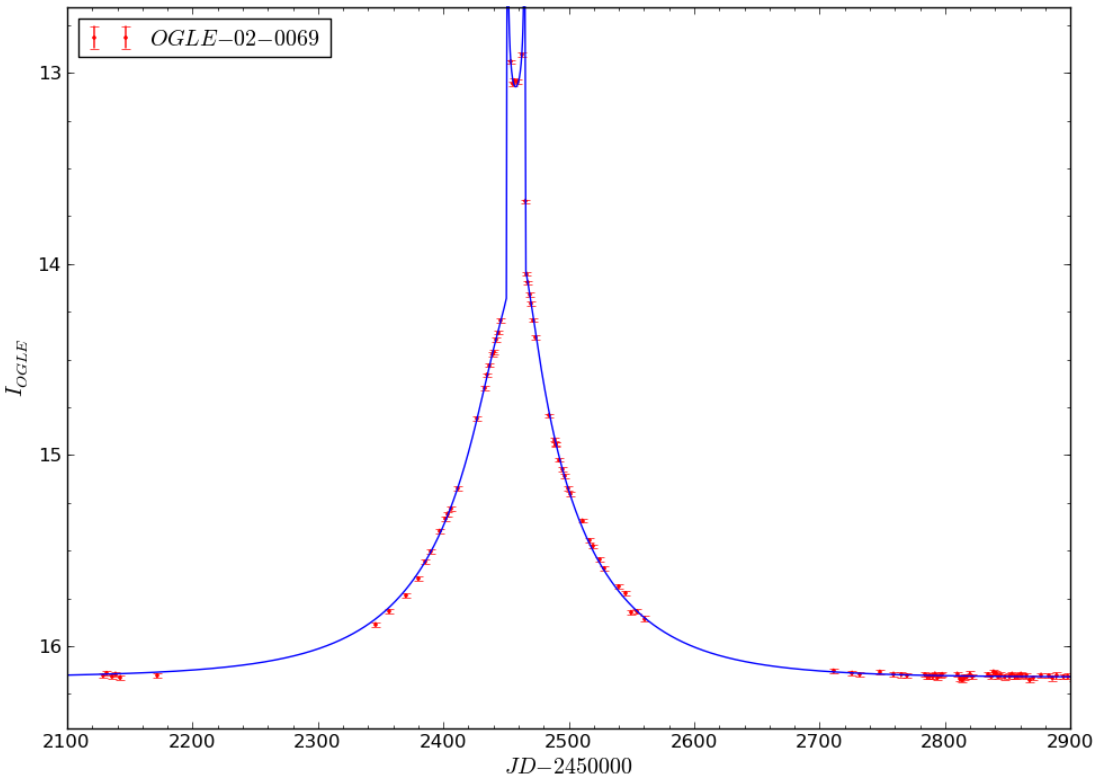


FIGURE 7.7: A light curve of the model solution (Table 7.2) of OGLE-2002-BLG-0069 determined by the GPU search method.

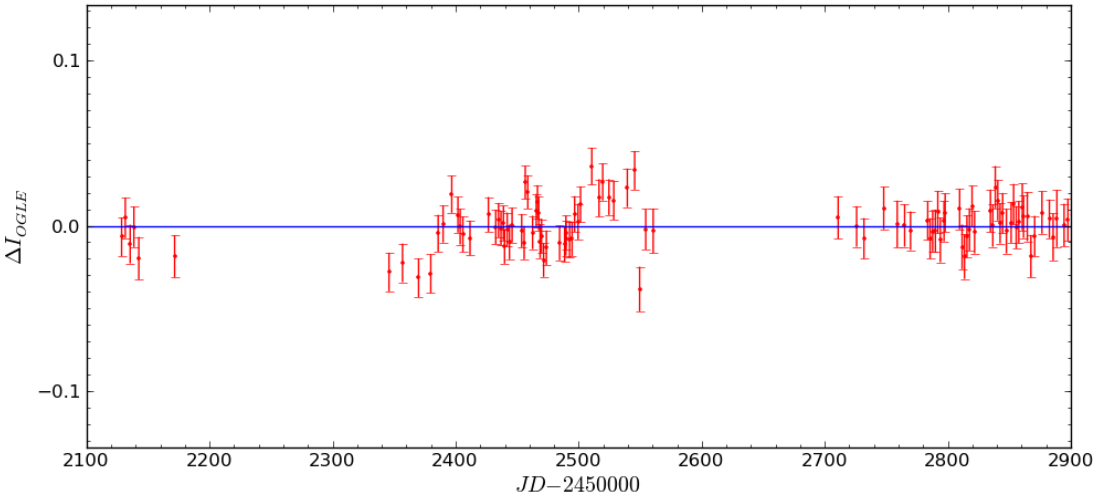


FIGURE 7.8: Light curve residuals of the model solution (Table 7.2) of OGLE-2002-BLG-0069 determined by the GPU search method.

M. Jaroszyński's solution (Table 7.2) at first appearance seems different to the minimized set presented in Table 7.2, which comes from being the degenerate close solution. The solution chosen by the GPU search method favored the wide solution, however, the close MCMC solutions all converged to a similar parameter set as M. Jaroszyński's, shown in Table 7.2, where a $\Delta\chi^2 \approx 8$, exists between the best GPU close and wide solutions.

7.2.3 OGLE-2002-BLG-0114

The modelling routine of this event was started by providing 46 data points between the AJDs of 2100.0 and 2550.0, requirements for the model to contain at least two peaks located near AJD 2400.79919 and 2416.9, and achieving a minimum magnitude change of at least 1.0. Analysing this event took 2 hours and 17 minutes, which performed a d, q grid search and MCMC search in areas of local minima (Figure 7.9). This resulted in the parameter set detailed in Table 7.3, where the errors are determined from an EMCEE search of an error-normalized data set. The binary lens solution produces a caustic structure, light curve, and residuals shown in Figures 7.10-7.12.

TABLE 7.3: Binary lens model solutions of OGLE-2002-BLG-0114, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.623	0.6529 ± 0.0028	1.54 ± 0.29
q	0.745	0.7451 ± 0.0021	2.5 ± 1.1
ρ	–	1.1×10^{-7}	0.14 ± 0.10
u_0	-0.04	-0.0188 ± 0.0019	-0.019 ± 0.059
ϕ	1.451	1.45266 ± 0.00042	0.27 ± 0.28
t_0	2412.5	2416.74 ± 0.28	2404.1 ± 2.1
t_E	75.8	78.03 ± 0.24	20.0 ± 9.3
χ^2	591.569	190.051	75.653

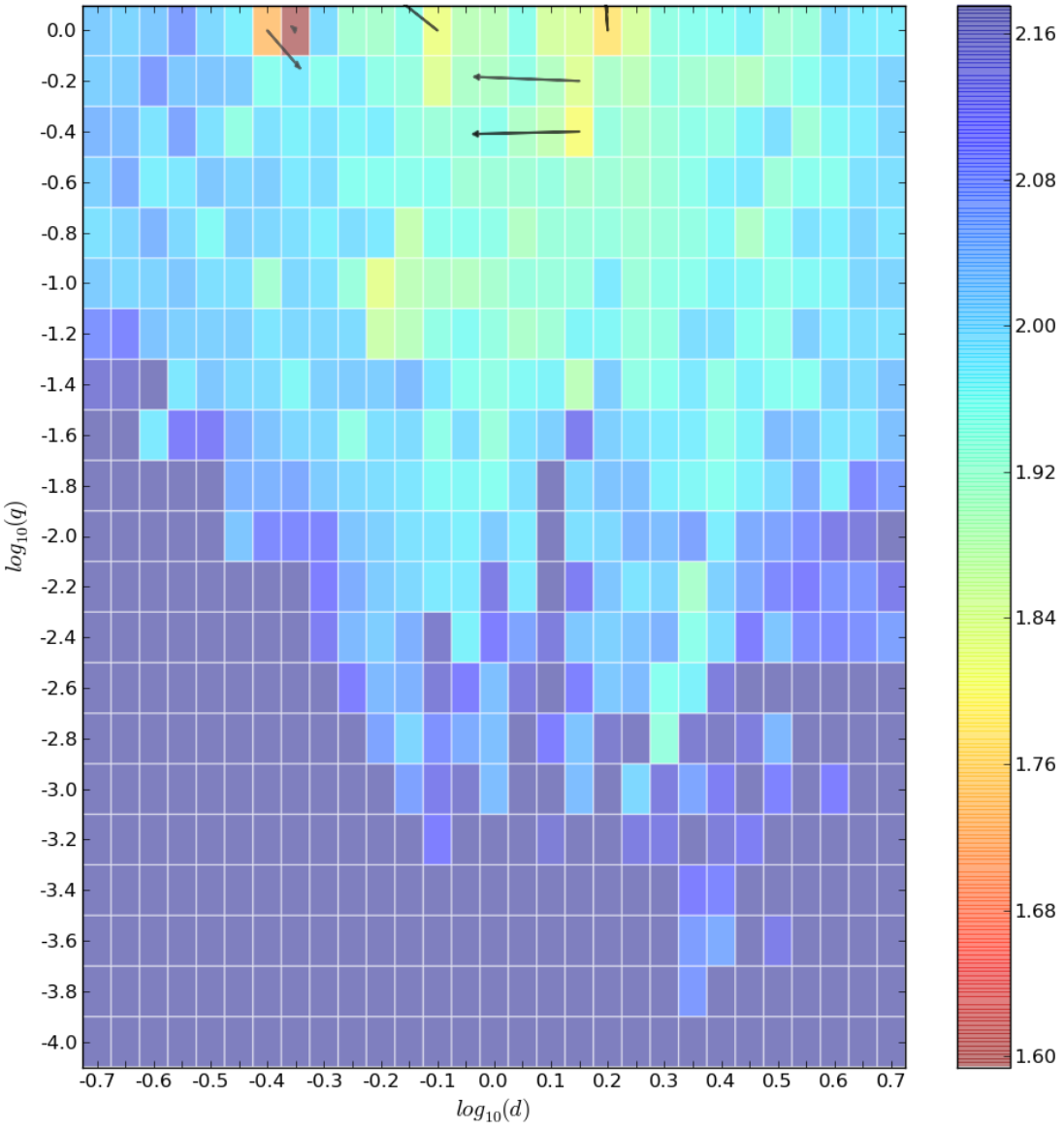


FIGURE 7.9: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2002-BLG-0114.

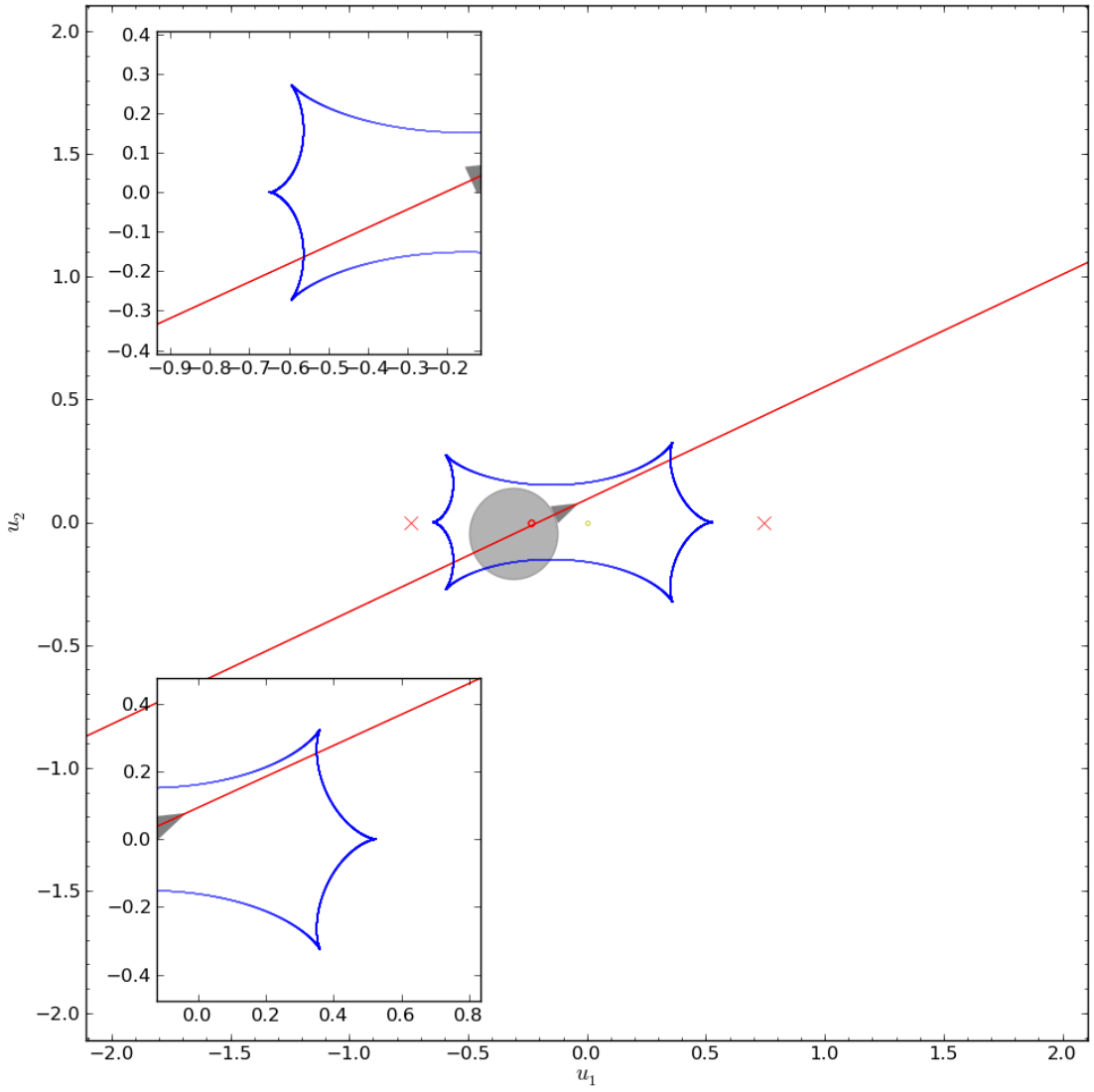


FIGURE 7.10: A caustic map with overlaid trajectory for the model solution (Table 7.3) of OGLE-2002-BLG-0114 determined by the GPU search method.

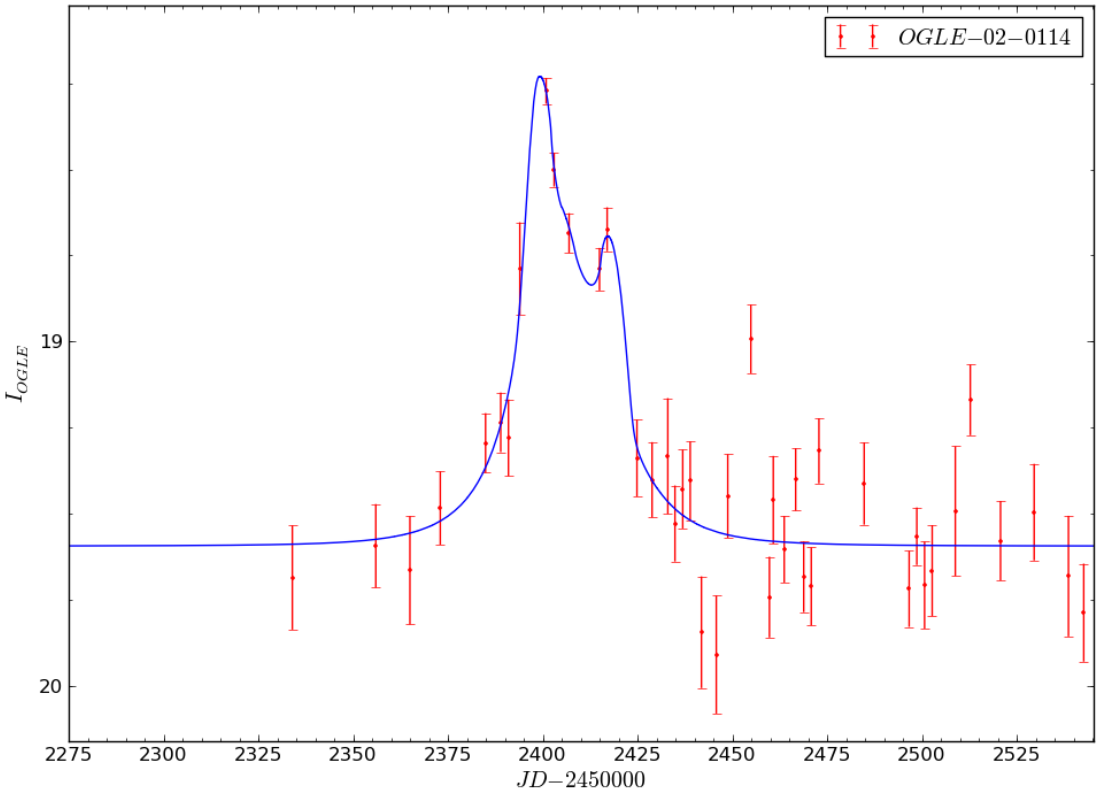


FIGURE 7.11: A light curve of the model solution (Table 7.3) of OGLE-2002-BLG-0114 determined by the GPU search method.

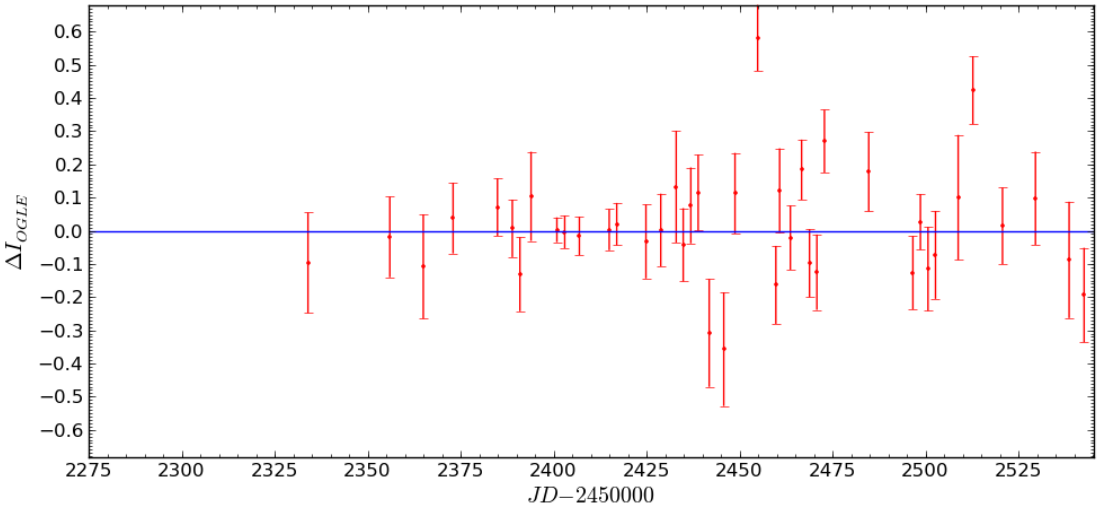


FIGURE 7.12: Light curve residuals of the model solution (Table 7.3) of OGLE-2002-BLG-0114 determined by the GPU search method.

M. Jaroszyński identifies an alternate solution (Table 7.3) to the one presented here. Using his solution a poor fitting model is produced, likely due to rounding in the parameter values and the sensitive nature of the data point that lies half way up the caustic entry. Therefore, an EMCEE minimization search method is performed around the area of his parameter solution. An exact match to his solution cannot be found, but the best parameter set located in the local area produced a comparable χ^2 solution. Despite the minimization exploration around this area the best solution was found to still be worse than the best close and wide solutions produced with the GPU method, by a $\Delta\chi^2$/DoF of over 2.8.

7.2.4 OGLE-2003-BLG-0021

To start the analysis of this event the modelling procedure was provided with the requirement that the model must contain at least two peaks with one being located around AJD of 2786.78185, and the model must achieve at least a minimum change in magnitude of 3.4. Using a total of 273 data points between the AJDs of 2000.0 and 3500.0, the search of all parameter space (Figure 7.13) was performed in 1 hour 56 minutes. This search found a global minimum with the parameter set shown in Table 7.4, which produce the caustic, light curve, and residuals Figures 7.14-7.16. To determine the error terms for each parameter, an additional EMCEE search was performed around this parameter set using error-normalized data.

TABLE 7.4: Binary lens model solutions of OGLE-2003-BLG-0021, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.941	0.9442 ± 0.0033	0.9468 ± 0.0049
q	0.799	0.800 ± 0.011	1.297 ± 0.062
ρ	–	1.1×10^{-7}	$3.86 \pm 0.34 \times 10^{-5}$
u_0	-0.09	-0.0849 ± 0.0034	-0.0767 ± 0.0050
ϕ	1.0107	1.0113 ± 0.0010	4.14 ± 0.19
t_0	2776.8	2776.816 ± 0.079	2776.5 ± 2.5
t_E	54.9	54.898 ± 0.073	55.6 ± 2.7
χ^2	10828.708	700.413	700.803

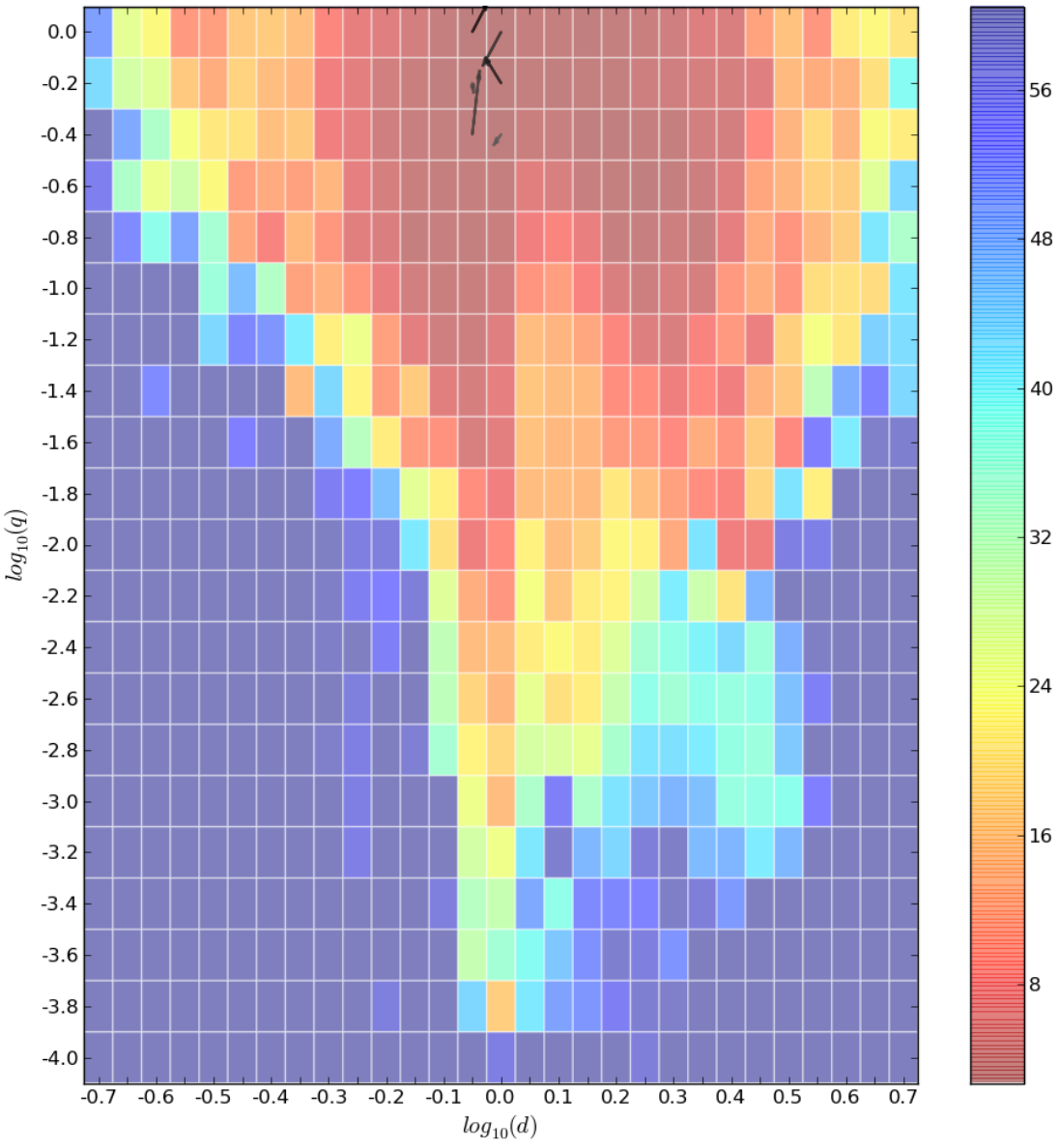


FIGURE 7.13: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0021.

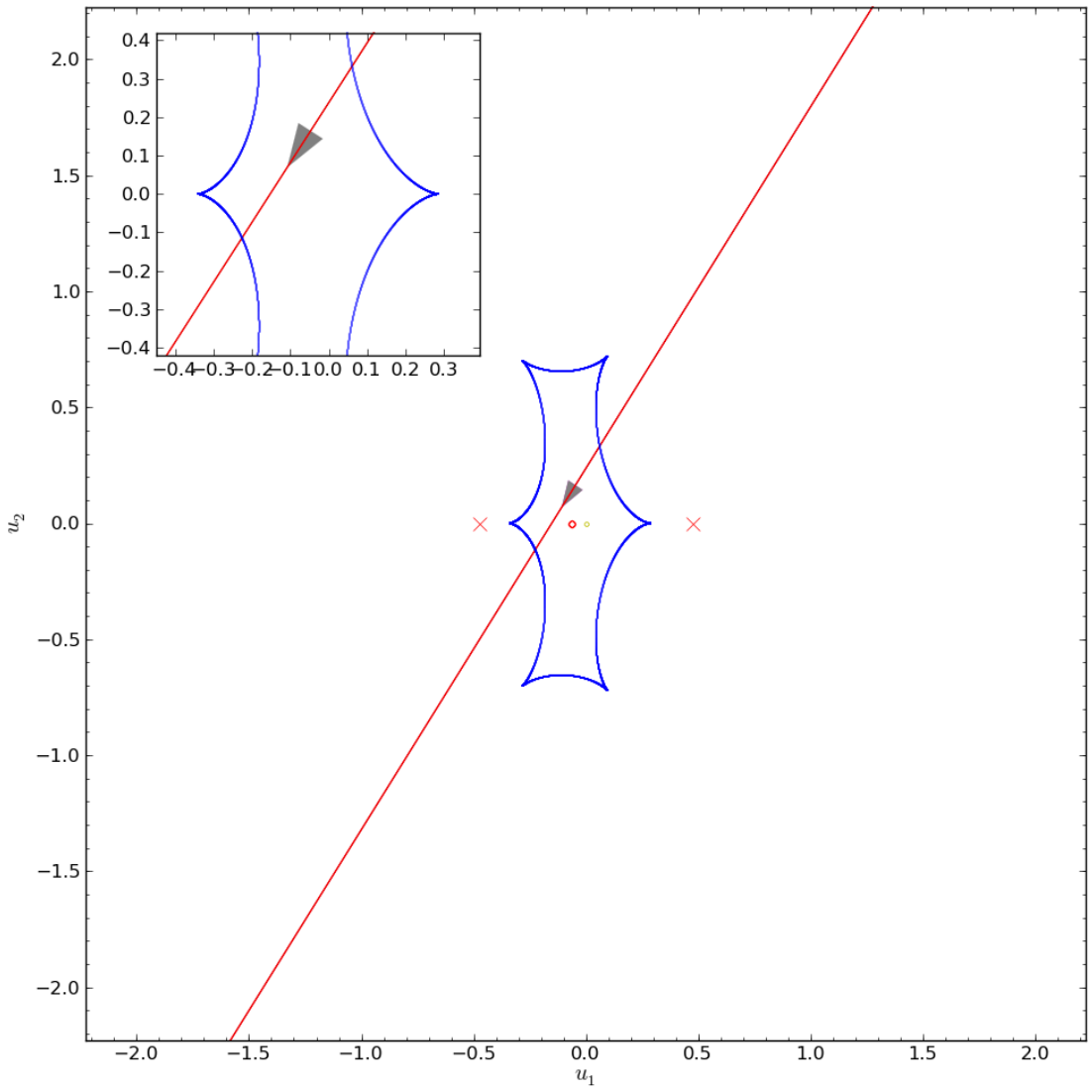


FIGURE 7.14: A caustic map with overlaid trajectory for the model solution (Table 7.4) of OGLE-2003-BLG-0021 determined by the GPU search method.

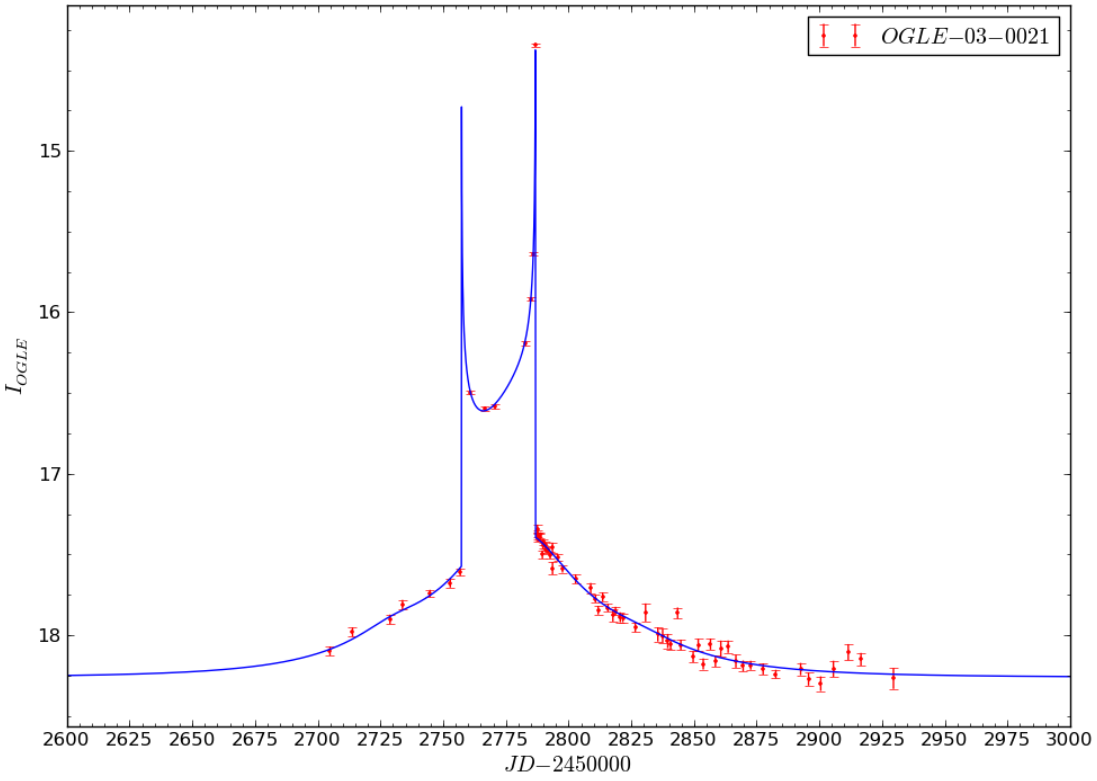


FIGURE 7.15: A light curve of the model solution (Table 7.4) of OGLE-2003-BLG-0021 determined by the GPU search method.

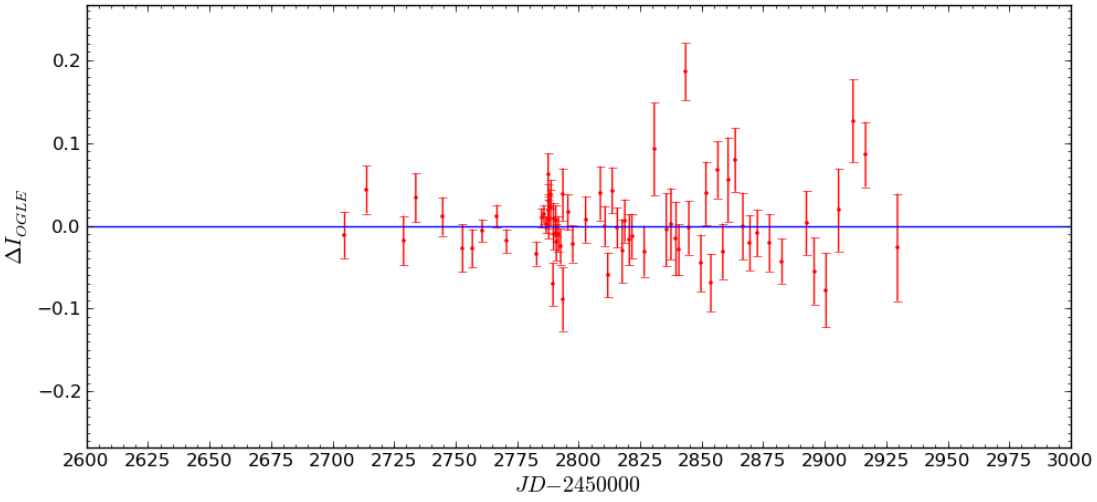


FIGURE 7.16: Light curve residuals of the model solution (Table 7.4) of OGLE-2003-BLG-0021 determined by the GPU serach method.

As shown in Table 7.4 the global minimum determined by the GPU search is found to be the same as M. Jaroszyński, although it appears slightly different as the GPU presented mass ratio (q) has reversed the lens masses. They can be shown to be equal by taking the mass ratio as $1/q$ and modifying u_0 and ϕ appropriately.

7.2.5 OGLE-2003-BLG-0056

Modelling of this event began with the requirement to determine a parameter set that produced a model with at least three peaks, where one of them is located close to AJD 2754.80676 and the model will achieve a change in magnitude of at least 1.25. Data was provided between the AJDs of 2000.0 and 3500.0 which contained 270 points. A search (Figure 7.17) was performed in 2 hours and 13 minutes locating the parameter set shown in Table 7.5, and Figures 7.18-7.20. This was followed by an EMCEE search of error-normalized data to determine the error terms for each parameter.

TABLE 7.5: Binary lens model solutions of OGLE-2003-BLG-0056, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	1.497	1.4967 ± 0.0018	1.180 ± 0.013
q	0.743	1.3459 ± 0.0019	0.310 ± 0.019
ρ	–	1.1×10^{-7}	0.0079 ± 0.0011
u_0	-0.02	-0.0082 ± 0.0053	-0.3033 ± 0.0057
ϕ	2.4211	5.56272 ± 0.00041	3.425 ± 0.134
t_0	2764.8	2764.80 ± 0.13	2758.45 ± 0.53
t_E	40.3	40.30 ± 0.16	46.25 ± 0.50
χ^2	938.617	929.465	1030.822

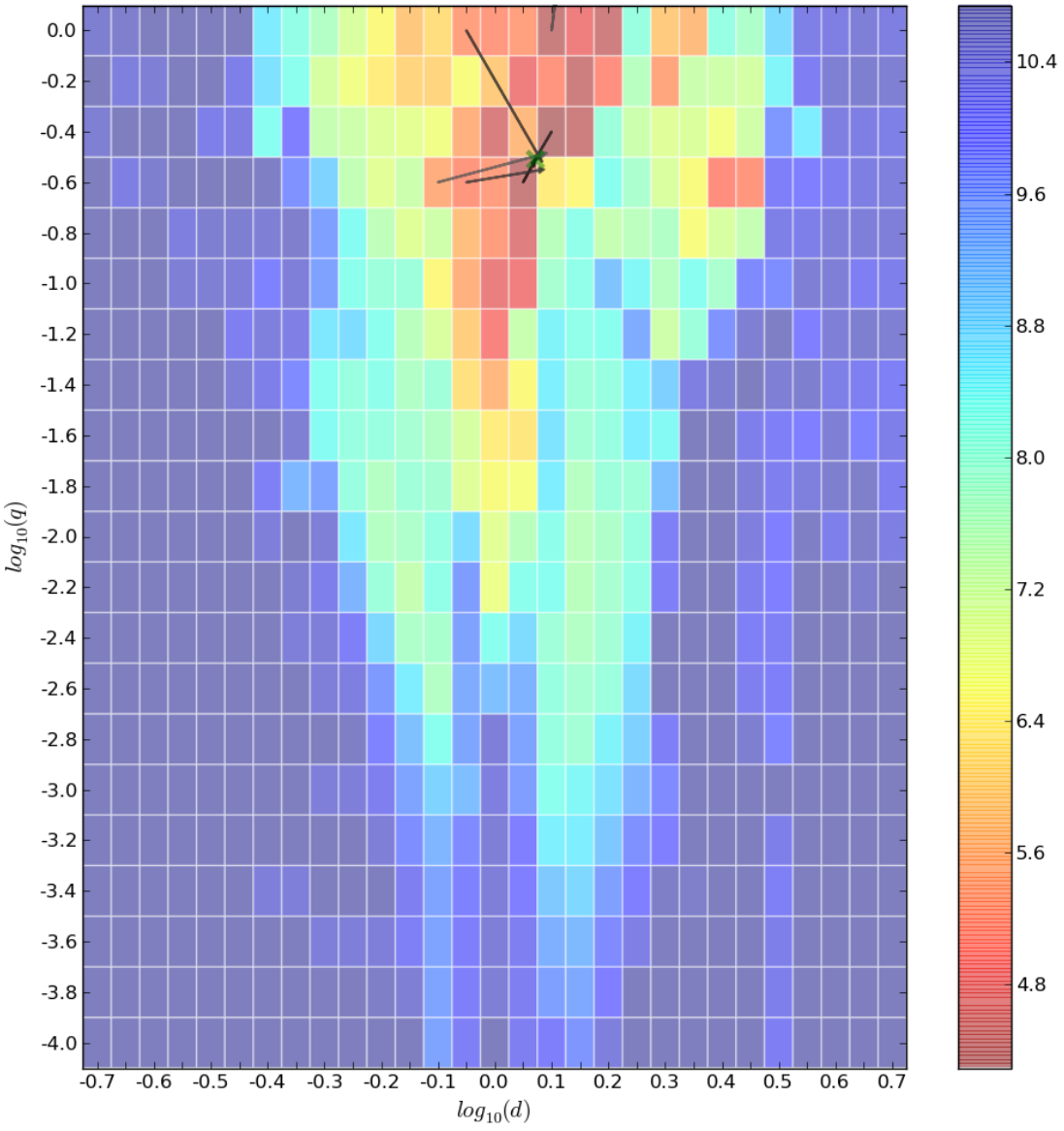


FIGURE 7.17: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0056.

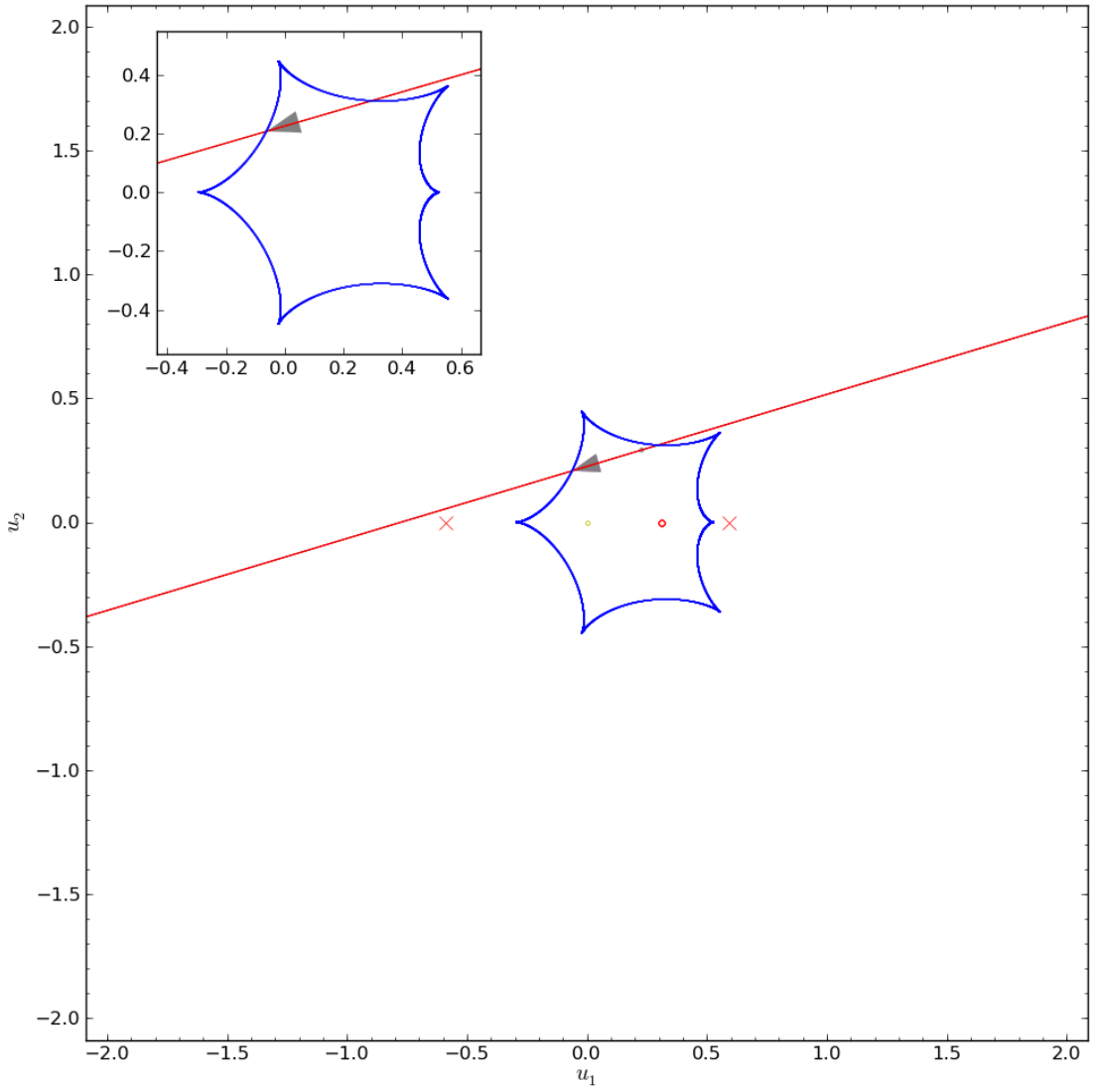


FIGURE 7.18: A caustic map with overlaid trajectory for the model solution (Table 7.5) of OGLE-2003-BLG-0056 determined by the GPU search method.

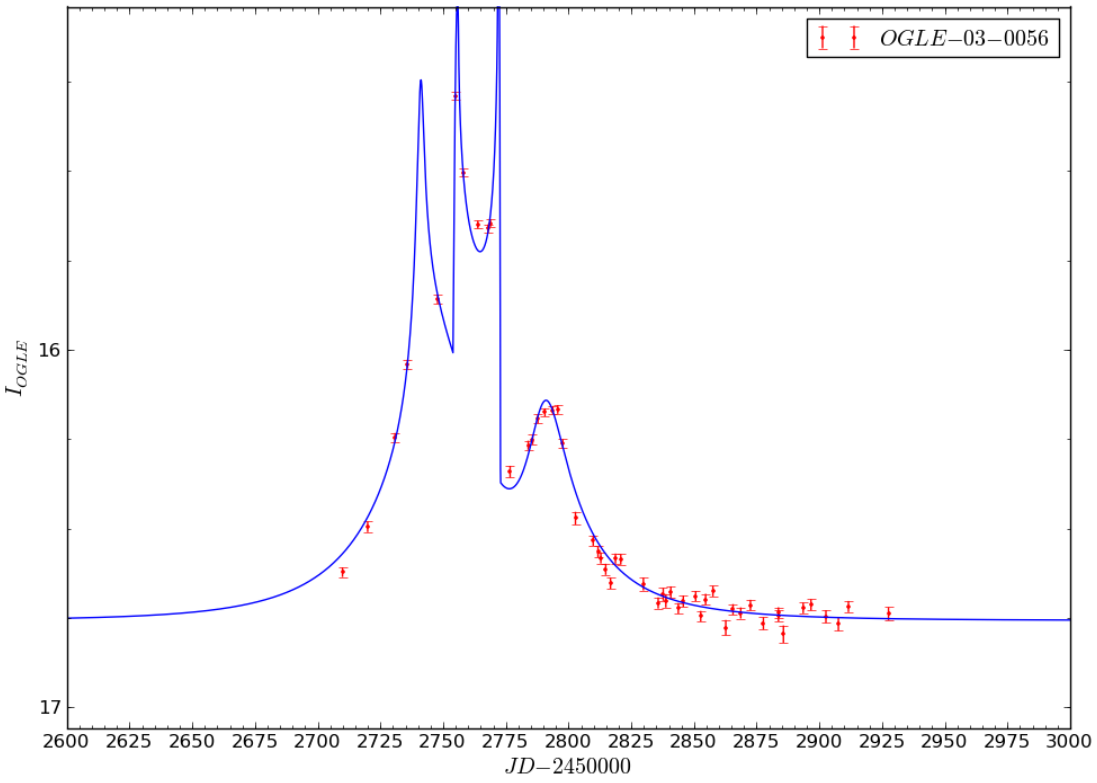


FIGURE 7.19: A light curve of the model solution (Table 7.5) of OGLE-2003-BLG-0056 determined by the GPU search method.

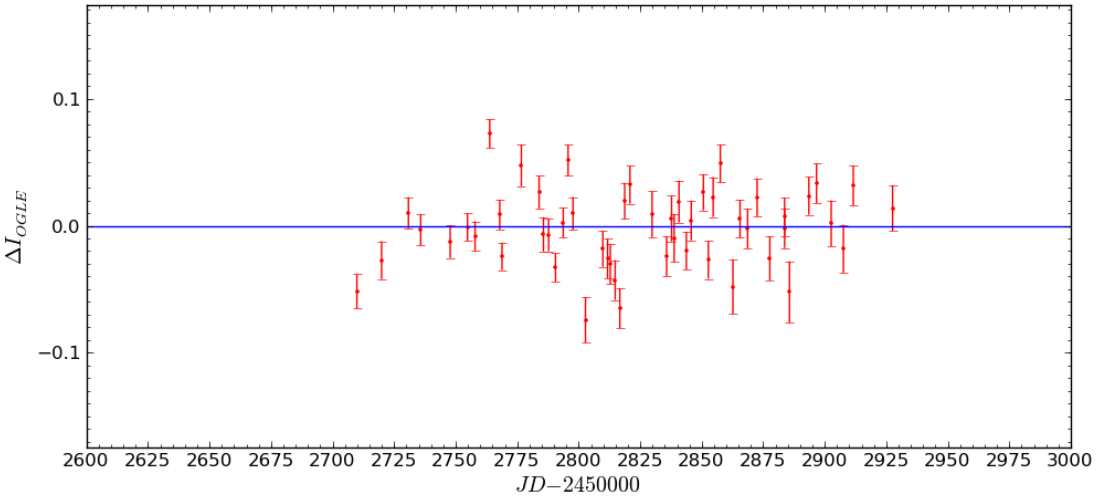


FIGURE 7.20: Light curve residuals of the model solution (Table 7.5) of OGLE-2003-BLG-0056 determined by the GPU search method.

Table 7.5 shows the parameter set determined by M. Jaroszyński, which produces a lower χ^2 solution by a $\Delta\chi^2/\text{DoF} \approx 0.4154$. On close inspection of the GPU modelling procedure, the initial d, q grid search successfully located this trajectory at its nearest d, q grid point. However, the minimized parameter set of this point was the fourth lowest pixel in the d, q map, but it was also the fourth lowest wide solution, by a $\Delta\chi^2$ of 3.26 meaning it was not selected, as only the best three wide solutions are chosen. This is an example of the limitations of a grid search method, where narrow valleys in the parameter space can fall between grid points. Indicating that to ensure no solutions are missed a larger search space is required (starting more local area MCMC searchs), however this comes at the added cost of a larger computing time.

7.2.6 OGLE-2003-BLG-0124

The modelling procedure for this event attempted to fit a model to 278 data points between the AJDs of 2000 and 3500. For a valid model it needed to achieve a minimum change in magnitude of at least 0.8 and contain two or more peaks, two of them being located around AJDs 2721.9 and 2769.7807. The search (Figure 7.21) took 2 hours and 24 minutes to complete, and identified the parameter set shown in Table 7.6, which produces Figures 7.22-7.24. An EMCEE search was carried out using an error-normalized set to identify the errors of each parameter.

TABLE 7.6: Binary lens model solutions of OGLE-2003-BLG-0124, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.959	0.9550 ± 0.0026	0.696 ± 0.077
q	0.666	0.6678 ± 0.0024	0.306 ± 0.099
ρ	–	1.1×10^{-7}	$2.8 \pm 2.4 \times 10^{-7}$
u_0	-0.11	-0.107 ± 0.072	-0.050 ± 0.021
ϕ	1.2666	1.26681 ± 0.00054	1.06 ± 0.10
t_0	2768.7	2769.15 ± 0.25	2762.2 ± 4.2
t_E	73.0	73.01 ± 0.29	306 ± 79
χ^2	611.641	611.454	614.126

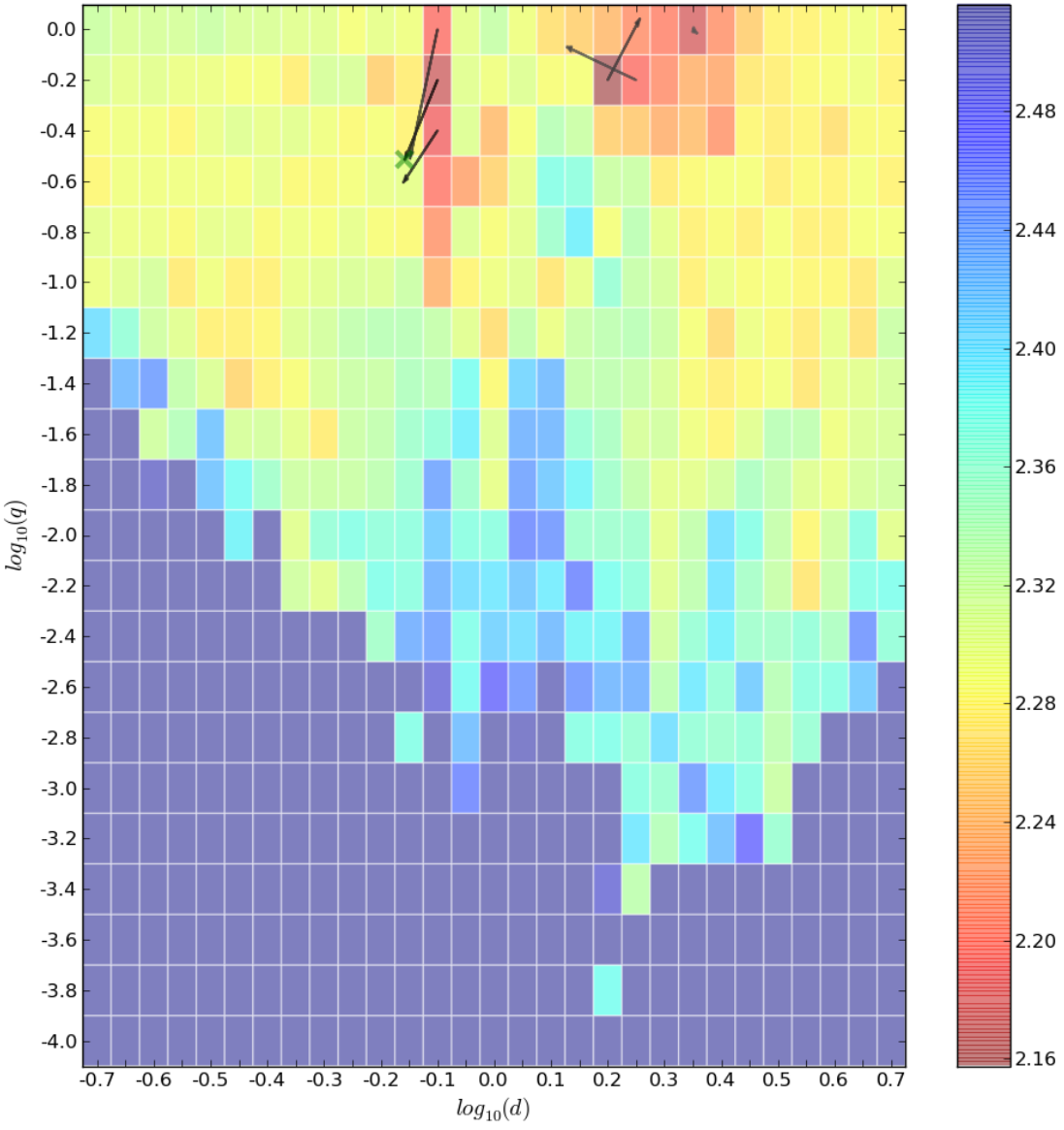


FIGURE 7.21: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0124.

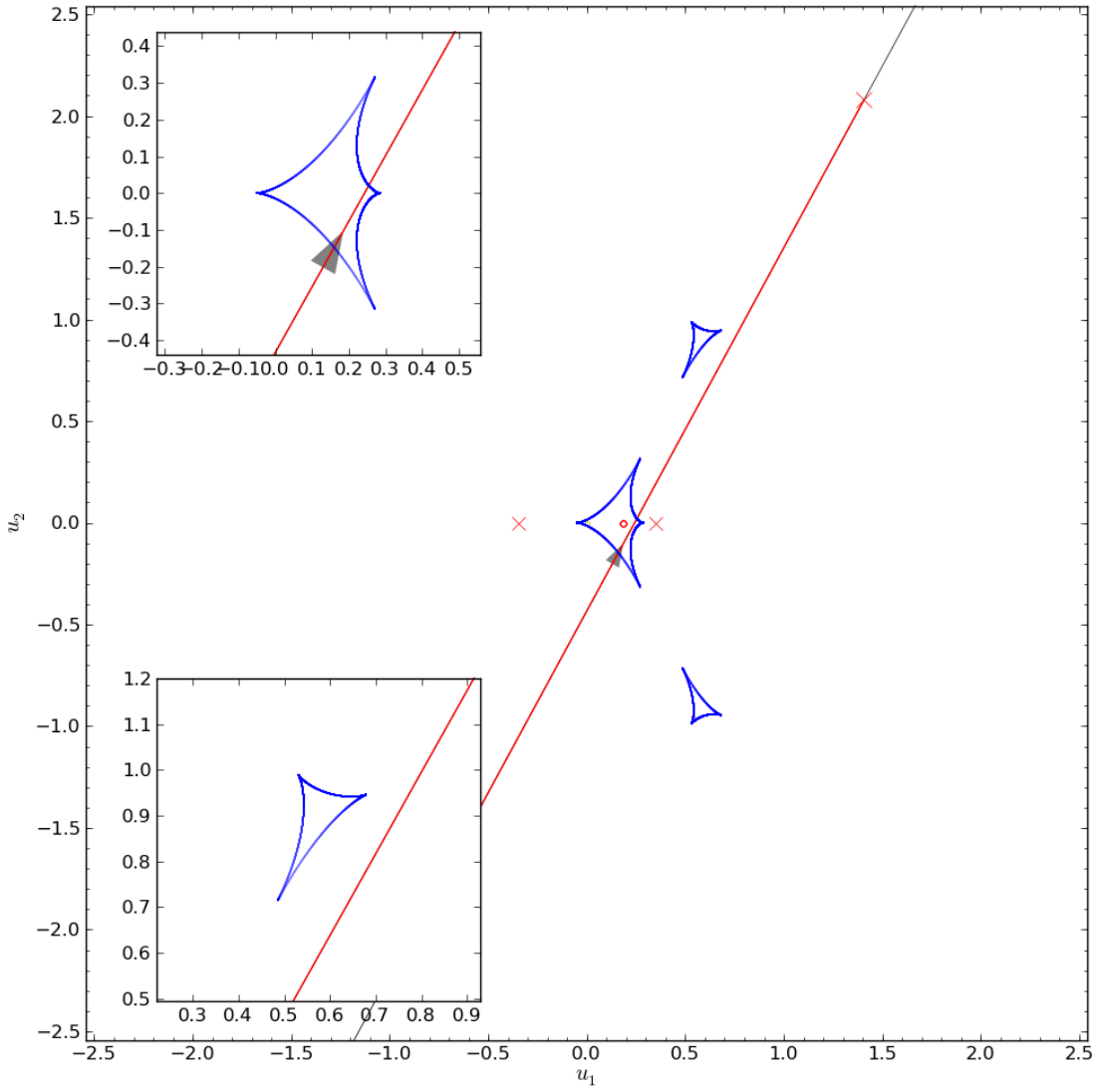


FIGURE 7.22: A caustic map with overlaid trajectory for the model solution (Table 7.6) of OGLE-2003-BLG-0124 determined by the GPU search method.

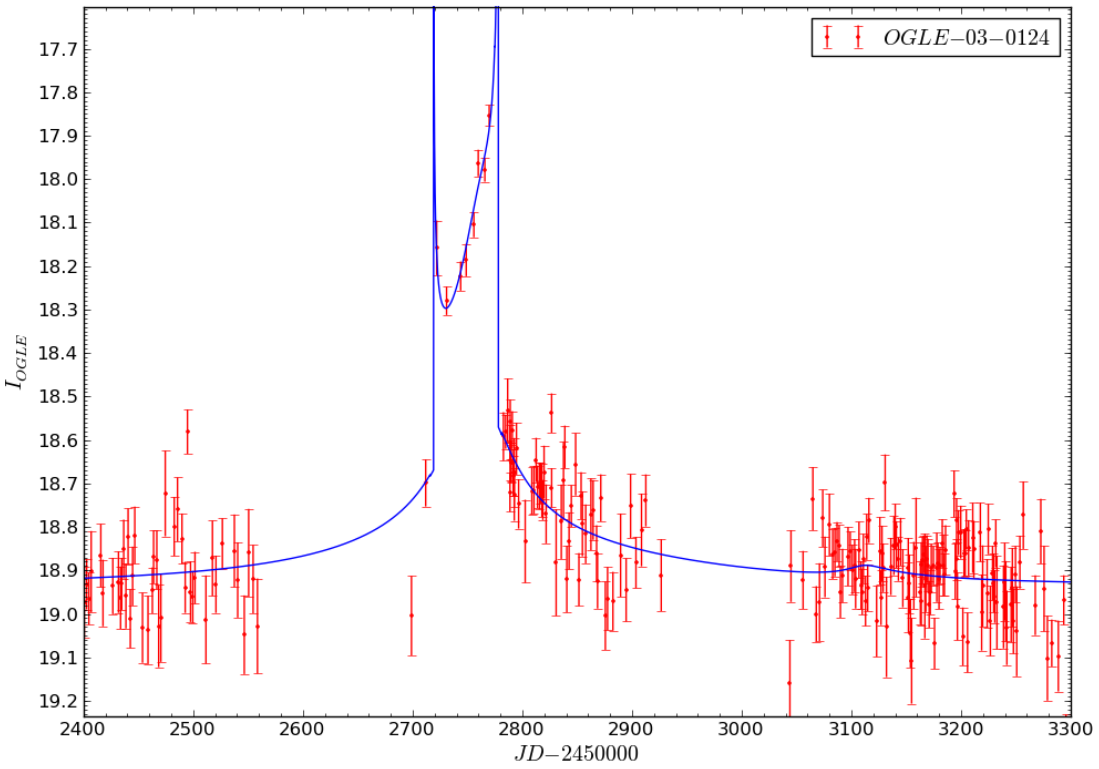


FIGURE 7.23: A light curve of the model solution (Table 7.6) of OGLE-2003-BLG-0124 determined by the GPU search method.

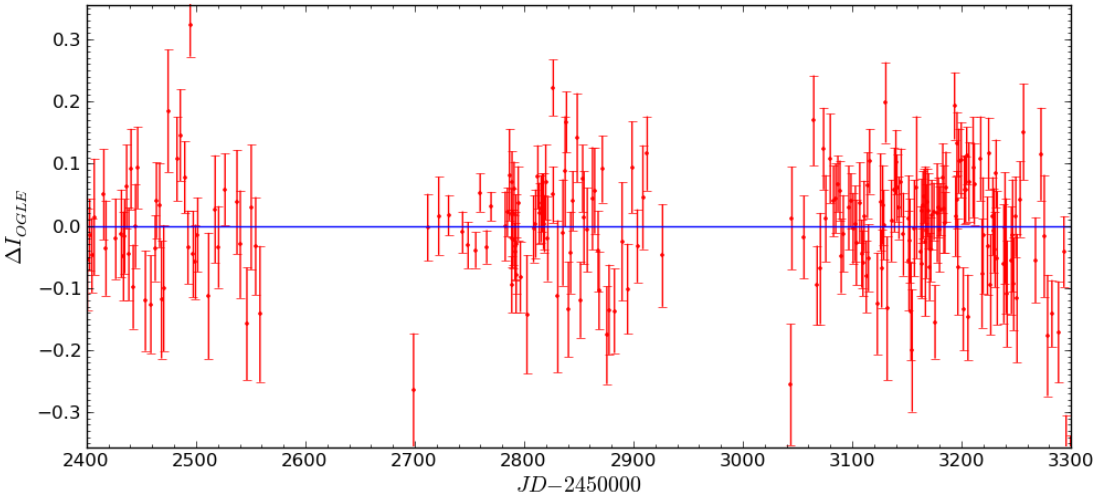


FIGURE 7.24: Light curve residuals of the model solution (Table 7.6) of OGLE-2003-BLG-0124 determined by the GPU search method.

The parameter set presented by M. Jaroszyński shown in Table 7.6 is different to the one determined by the GPU search method. On further analysis it is found that the largest contribution to difference in the model's χ^2 values do not come from the data over the caustic crossings (shown in Figures 7.25-7.27). Although M. Jaroszyński's model has a slightly lower χ^2/DoF of 0.026, the contributing factor to this difference comes from the low magnified data which has a lot of scatter just after the caustic exit. This shows that the global minimum area is not well defined by the caustic crossing and several models can exist that describe the event.

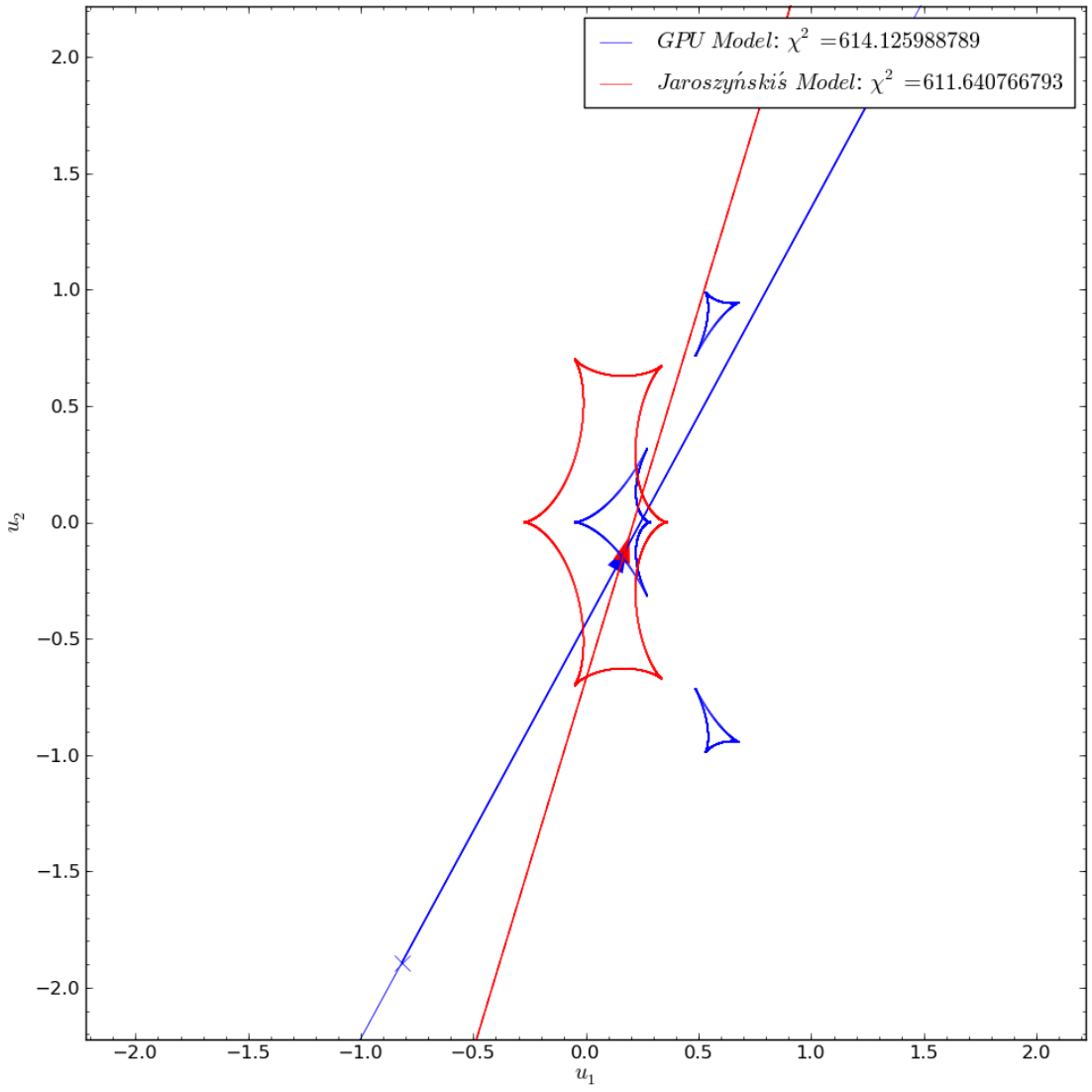


FIGURE 7.25: Caustic maps with overlaid trajectory of the GPU (blue) and M. Jaroszyński's (red) solutions.

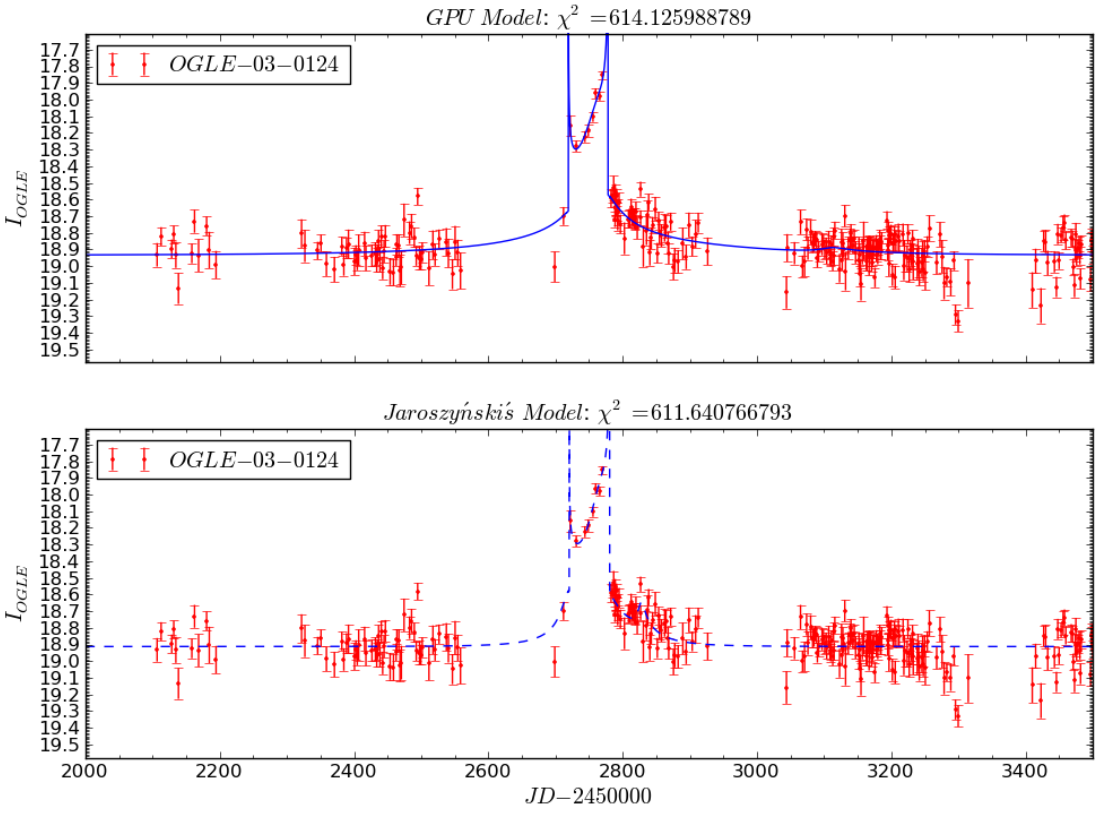


FIGURE 7.26: Light curve comparison of the GPU and M. Jaroszyński's solutions.

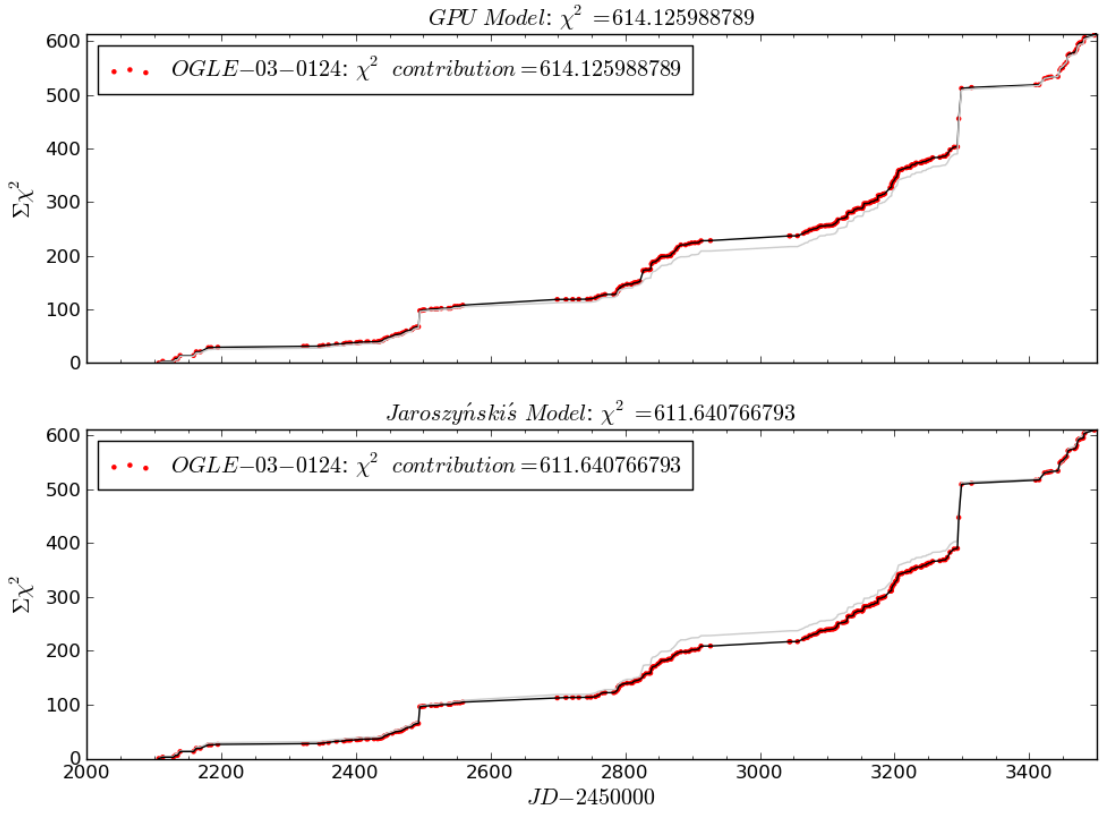


FIGURE 7.27: Cumulative χ^2 plots of the GPU and M. Jaroszyński's models (black line), with that comparison cumulative plot in grey.

7.2.7 OGLE-2003-BLG-0135

Modelling of this event was provided with the requirements that for a valid solution the model must contain at least two peaks located around the AJDs of 2767.78 and 2786.8, and a minimum change in magnitude of 0.8. The model was fit to 259 data points between the AJDs of 2000.0 and 3500.0. A grid search and six local MCMC explorations (Figure 7.28) took a total of 2 hours and 24 minutes to identify the global minimum, which was followed by an EMCEE search on error-normalized data to calculate the parameter's errors. The result of this search is shown in Table 7.7, which produce the caustic structure and trajectory, light curve, and residuals of Figures 7.29-7.31.

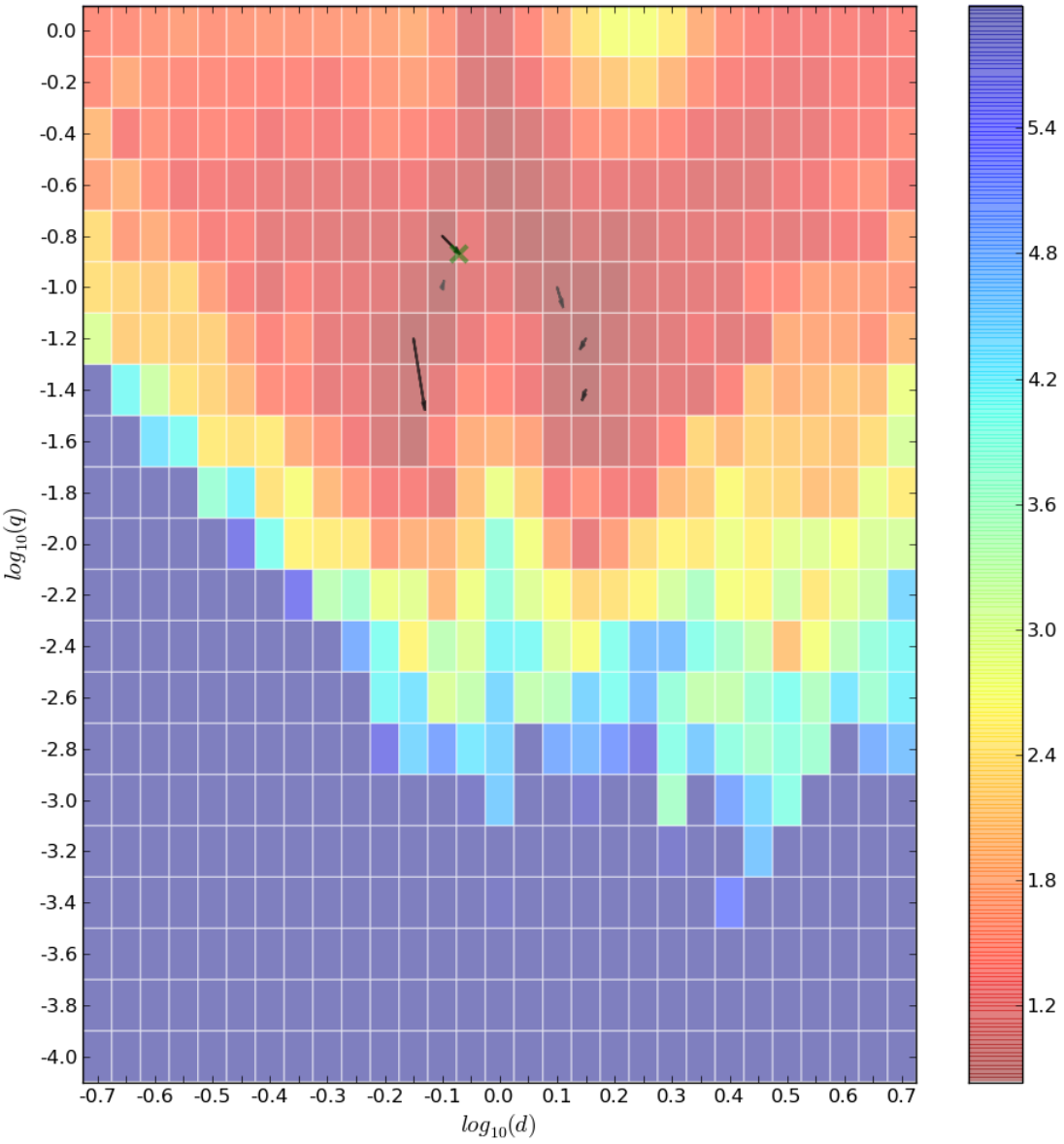


FIGURE 7.28: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0135.

TABLE 7.7: Binary lens model solutions of OGLE-2003-BLG-0135, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.847	0.84577 ± 0.00072	0.853 ± 0.031
q	0.129	0.13025 ± 0.00088	0.1384 ± 0.0072
ρ	–	1.1×10^{-7}	0.00345 ± 0.00019
u_0	-0.20	-0.1996 ± 0.0013	-0.216 ± 0.011
ϕ	2.1468	2.148 ± 0.00038	2.184 ± 0.039
t_0	2724.3	2724.362 ± 0.024	2726.7 ± 5.0
t_E	339.9	339.920 ± 0.021	324 ± 12
χ^2	206.911	195.474	189.801

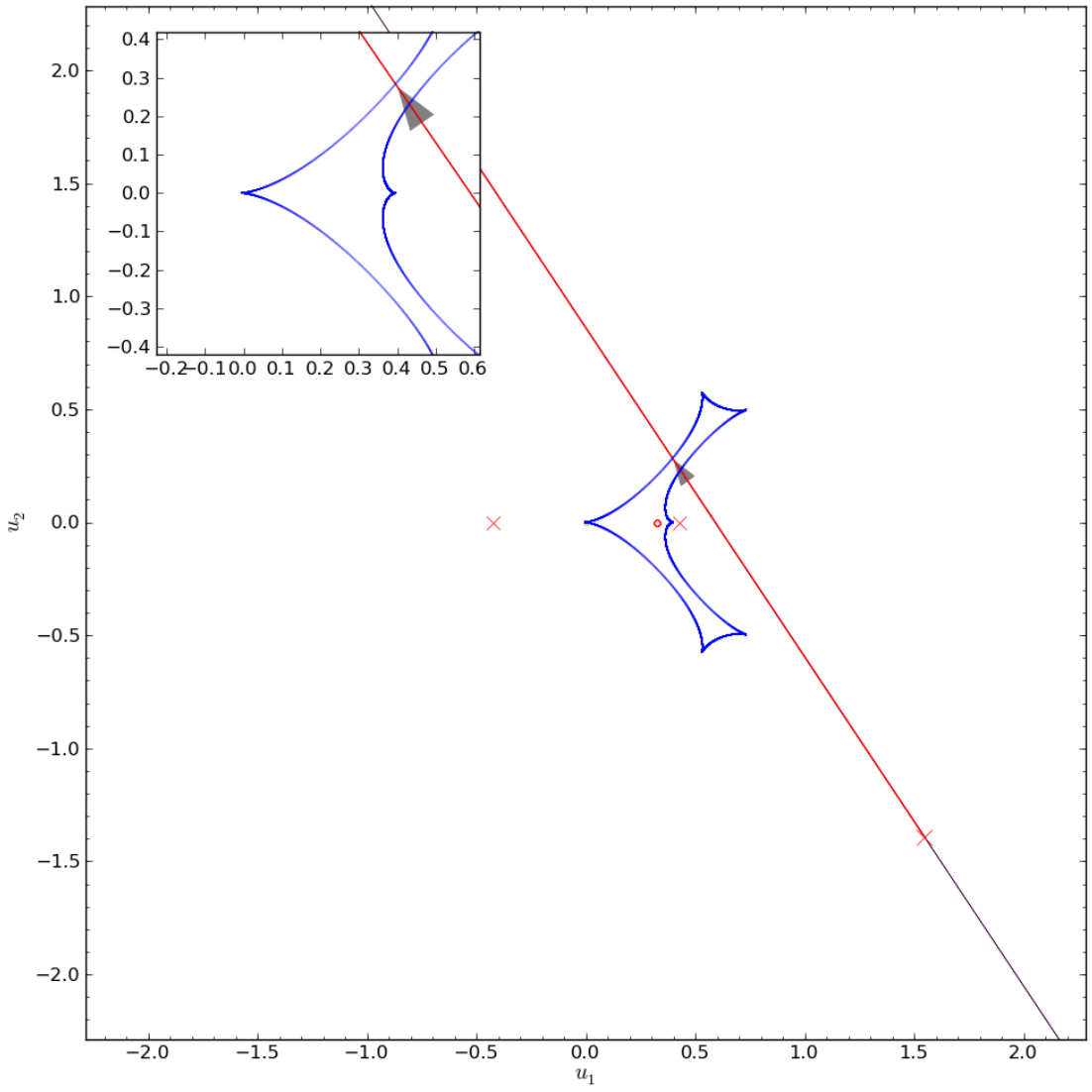


FIGURE 7.29: A caustic map with overlaid trajectory for the model solution (Table 7.7) of OGLE-2003-BLG-0135 determined by the GPU search method.

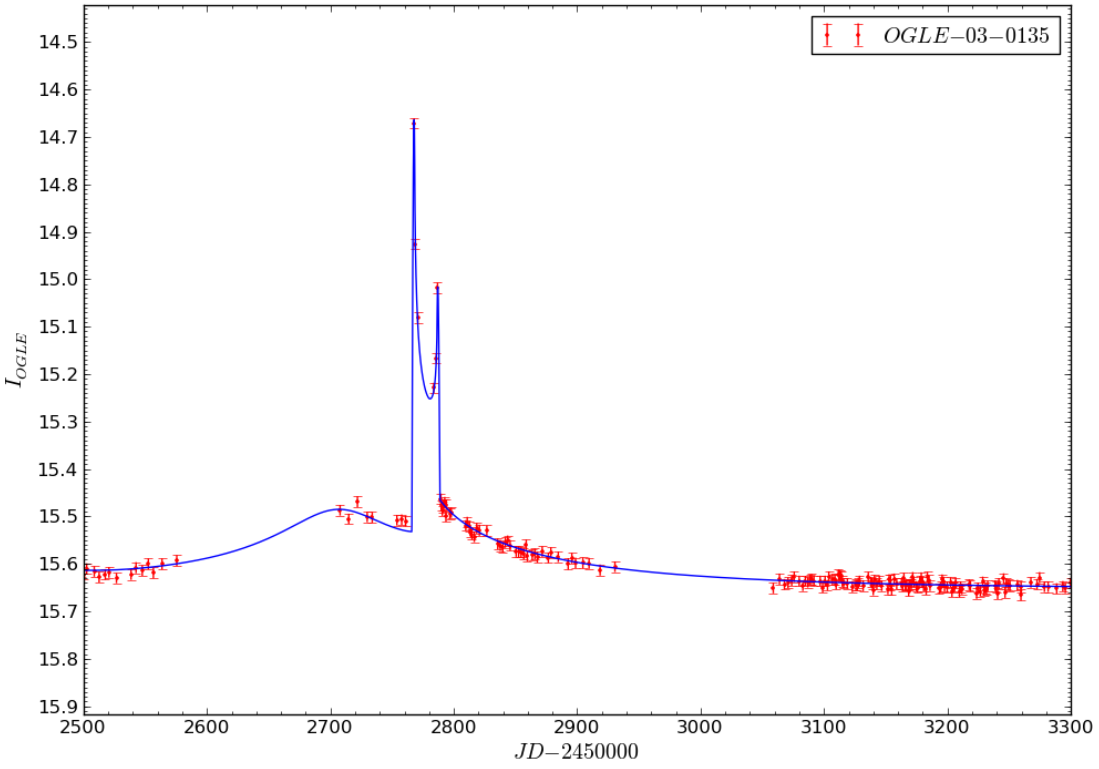


FIGURE 7.30: A light curve of the model solution (Table 7.7) of OGLE-2003-BLG-0135 determined by the GPU search method.

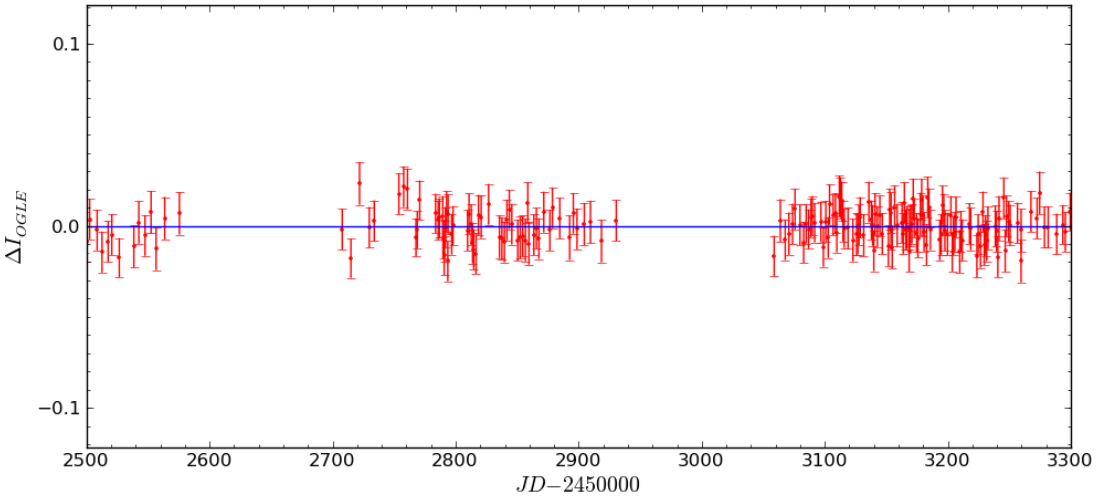


FIGURE 7.31: Light curve residuals of the model solution (Table 7.7) of OGLE-2003-BLG-0135 determined by the GPU search method.

The solution set presented by M. Jaroszyński, shown in Table 7.7, is located in the same area of global minimum as the GPU solution, with only small differences in the parameter values.

7.2.8 OGLE-2003-BLG-0170

The modelling routine of this event was initialized with 139 data points between the AJDs of 2400.0 and 3000.0, it required the solution to contain at least three peaks, with two being located near 2793.91355 and 2796.7985. The solution is also required to achieve a minimum magnitude change of at least 2.75. Analysing this event took 2 hours and 41 minutes to perform a d, q grid search and MCMC search in areas of local minima (Figure 7.32). The resulting parameter set is shown in Table 7.8, were the errors are determined from an EMCEE search of an error-normalized data set. These binary lens parameters produce a caustic structure, light curve, and residuals shown in Figures 7.33-7.35.

TABLE 7.8: Binary lens model solutions of OGLE-2003-BLG-0170, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	1.213	1.2162 ± 0.0019	0.834 ± 0.039
q	0.789	0.7898 ± 0.0017	0.225 ± 0.012
ρ	0.0027	0.00428 ± 0.00085	0.0025 ± 0.00012
u_0	-0.35	-0.3530 ± 0.0013	-0.1400 ± 0.0066
ϕ	2.3328	2.33200 ± 0.00031	2.63 ± 0.10
t_0	2794.1	2794.068 ± 0.023	2793.47 ± 0.69
t_E	15.6	15.637 ± 0.014	23.5 ± 1.2
χ^2	53745.368	5880.630	1489.322

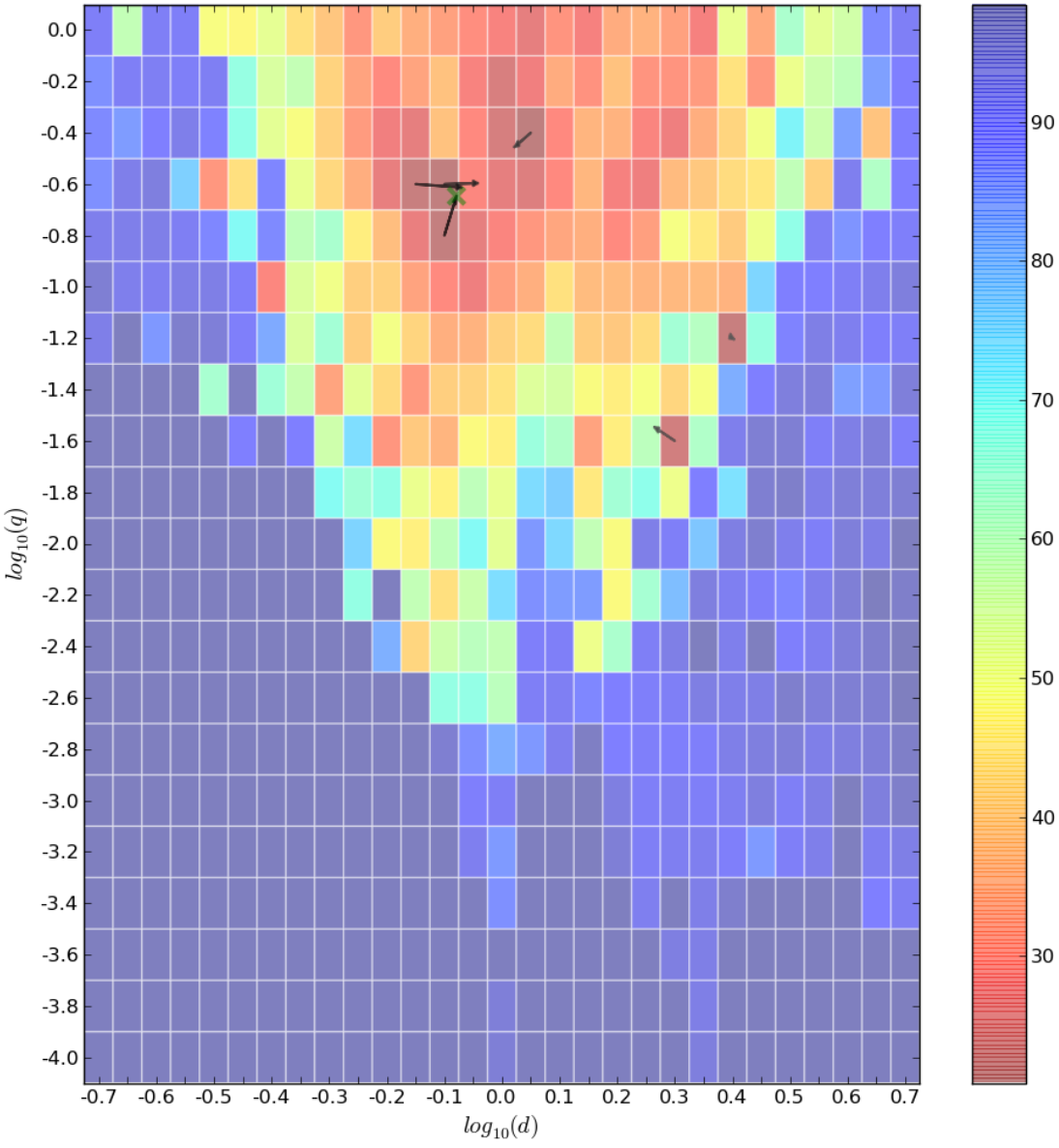


FIGURE 7.32: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0170.

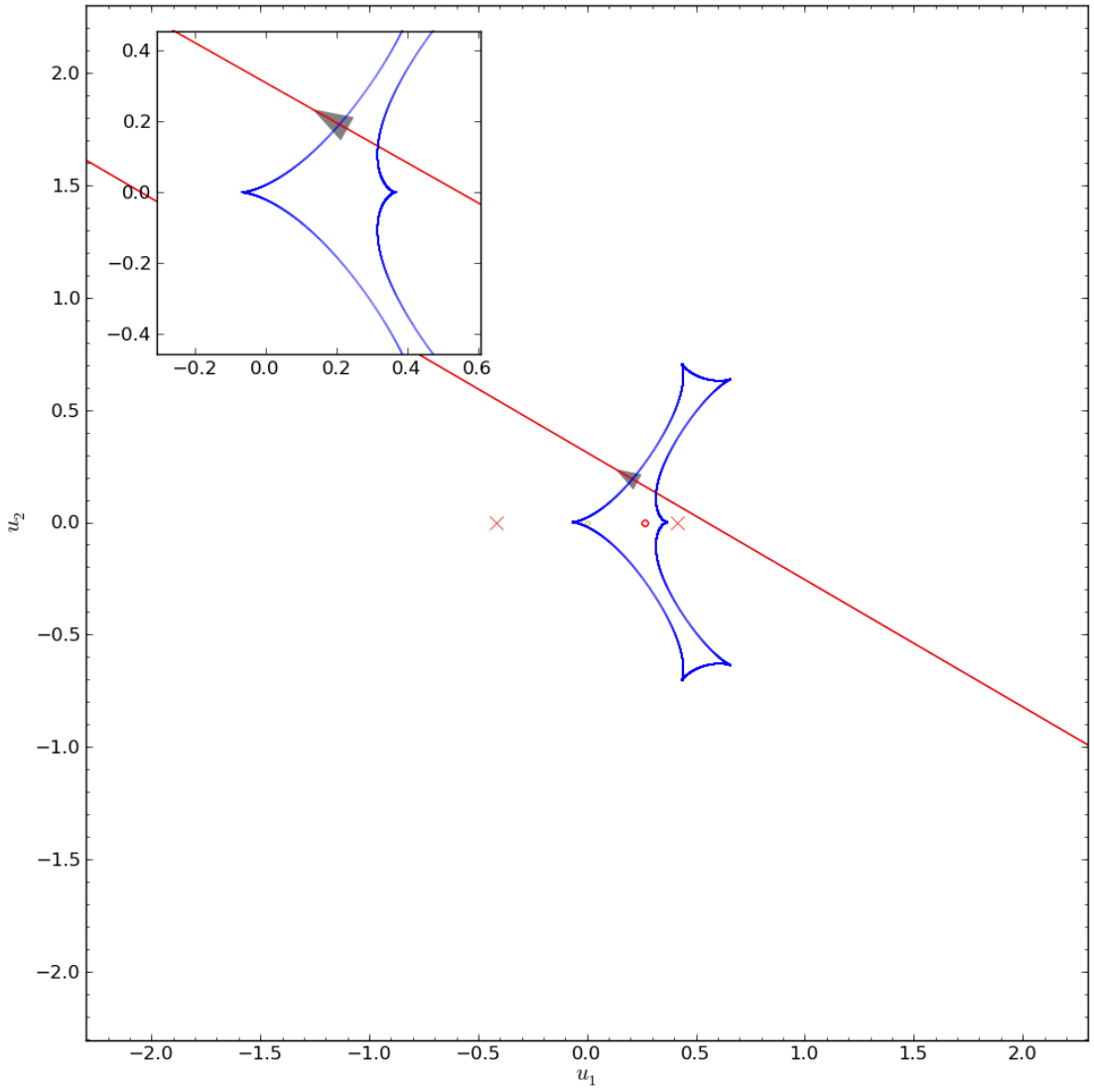


FIGURE 7.33: A caustic map with overlaid trajectory for the model solution (Table 7.8) of OGLE-2003-BLG-0170 determined by the GPU search method

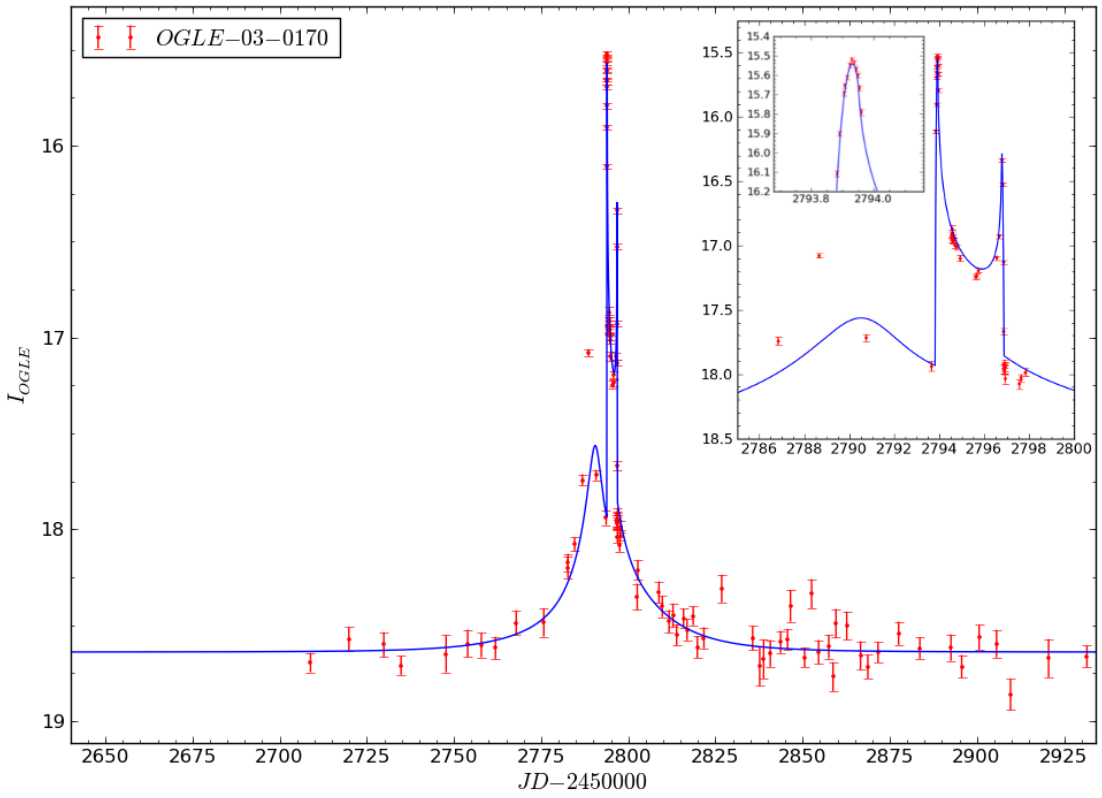


FIGURE 7.34: A light curve of the model solution (Table 7.8) of OGLE-2003-BLG-0170 determined by the GPU search method.

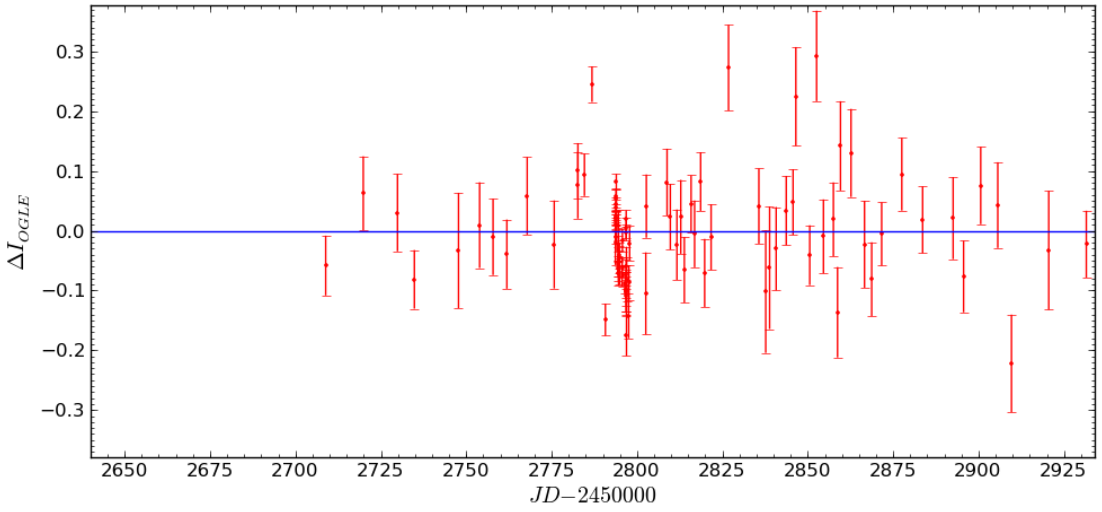


FIGURE 7.35: Light curve residuals of the model solution (Table 7.8) of OGLE-2003-BLG-0170 determined by the GPU search method.

M. Jaroszyński's parameter set (Table 7.8) is very different to the one found and presented here. Due to the densely observed caustic crossings the smallest of inaccuracies in an input parameter set can lead to very large χ^2 values. Despite using a parameter minimization method starting at M. Jaroszyński's solution, no suitable parameter set was found. As can be seen in Figure 7.34, the parameter set determined by the GPU search method is not ideal, and struggles to fit the initial peak (at AJD 2790). With a thoroughly observed caustic crossing, the χ^2 space becomes very sharp, making it challenging for minimization methods to locate narrow local valleys in the parameter space. To avoid this, a denser search of the parameter space is required, but this comes with heavy time penalties.

7.2.9 OGLE-2003-BLG-0200

This event was modeled with a requirement for valid models to achieve a minimum change in magnitude of 1.5, and contain at least two peaks, with one being located near the AJD of 2839.59704. The data used for the modelling process included 114 points between the AJDs of 2000.0-3500.0. A grid search and exploration with MCMC of six local minima (Figure 7.36) took 2 hours and 26 minutes to complete, before an EMCEE search of the global minimum was performed to determine the error terms for each parameter, detailed in Table 7.9. This produces the model's caustic, light curve, and residuals shown in Figures 7.37-7.39.

TABLE 7.9: Binary lens model solutions of OGLE-2003-BLG-0200, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	1.495	1.4960 ± 0.0013	0.654 ± 0.012
q	0.209	0.2093 ± 0.0022	0.180 ± 0.021
ρ	–	1.1×10^{-7}	0.0168 ± 0.0018
u_0	-0.06	-0.0593 ± 0.0010	0.0215 ± 0.0032
ϕ	2.1415	2.14155 ± 0.00028	2.052 ± 0.038
t_0	2836.4	2836.412 ± 0.020	2835.33 ± 0.16
t_E	46.0	46.025 ± 0.019	59.9 ± 3.7
χ^2	17903.545	186.443	186.596

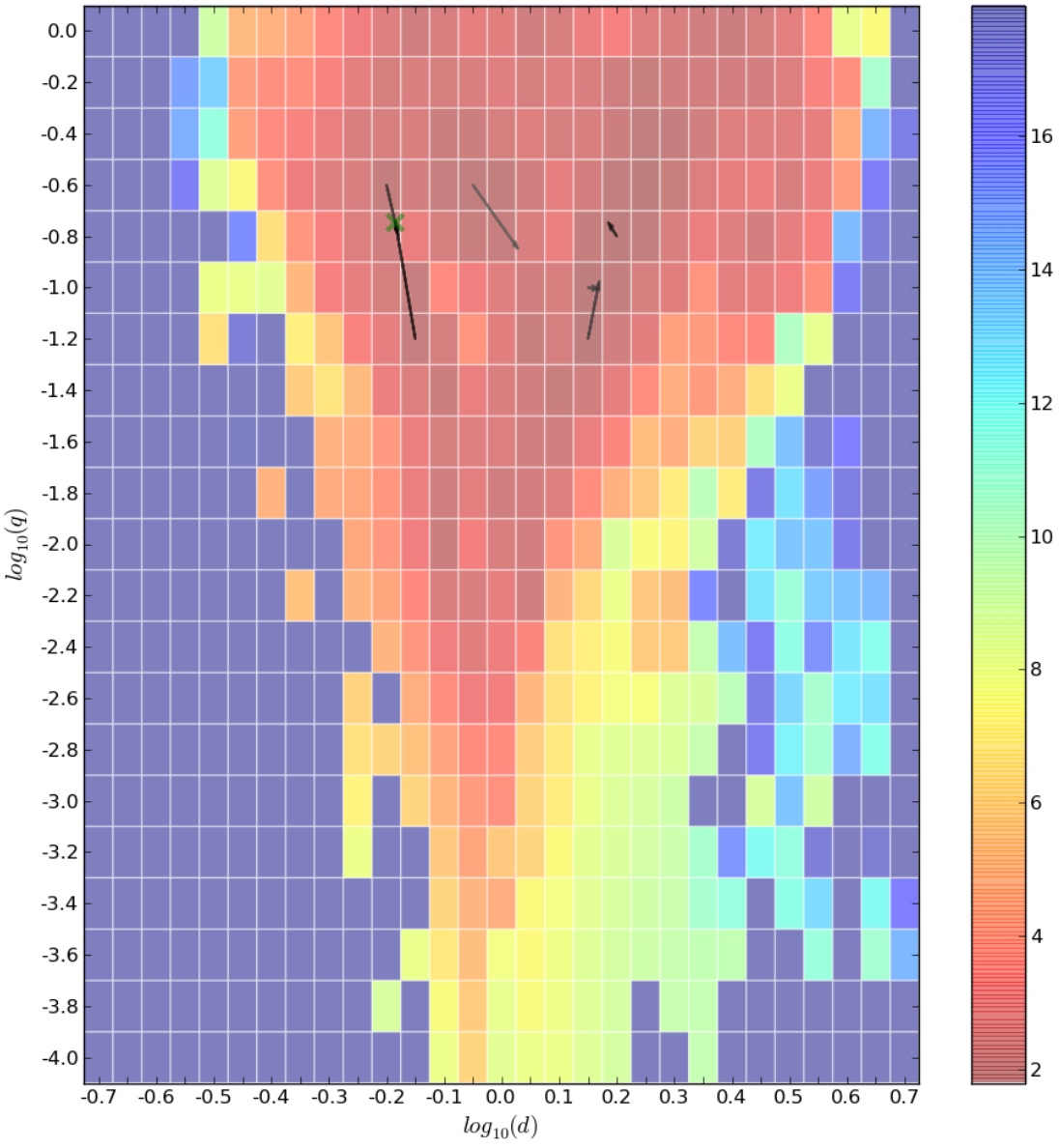


FIGURE 7.36: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0200.

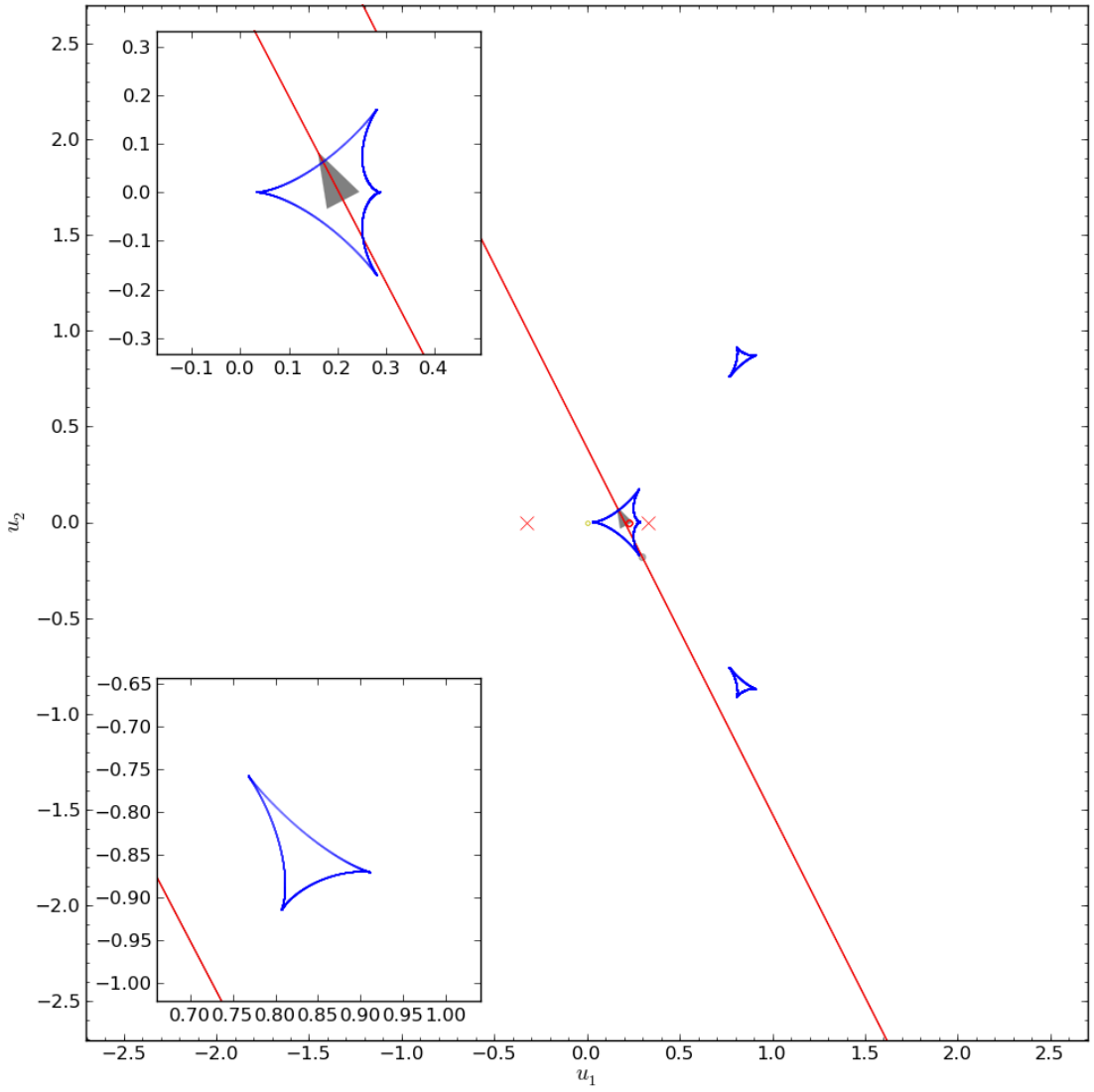


FIGURE 7.37: A caustic map with overlaid trajectory for the model solution (Table 7.9) of OGLE-2003-BLG-0200 determined by the GPU search method.

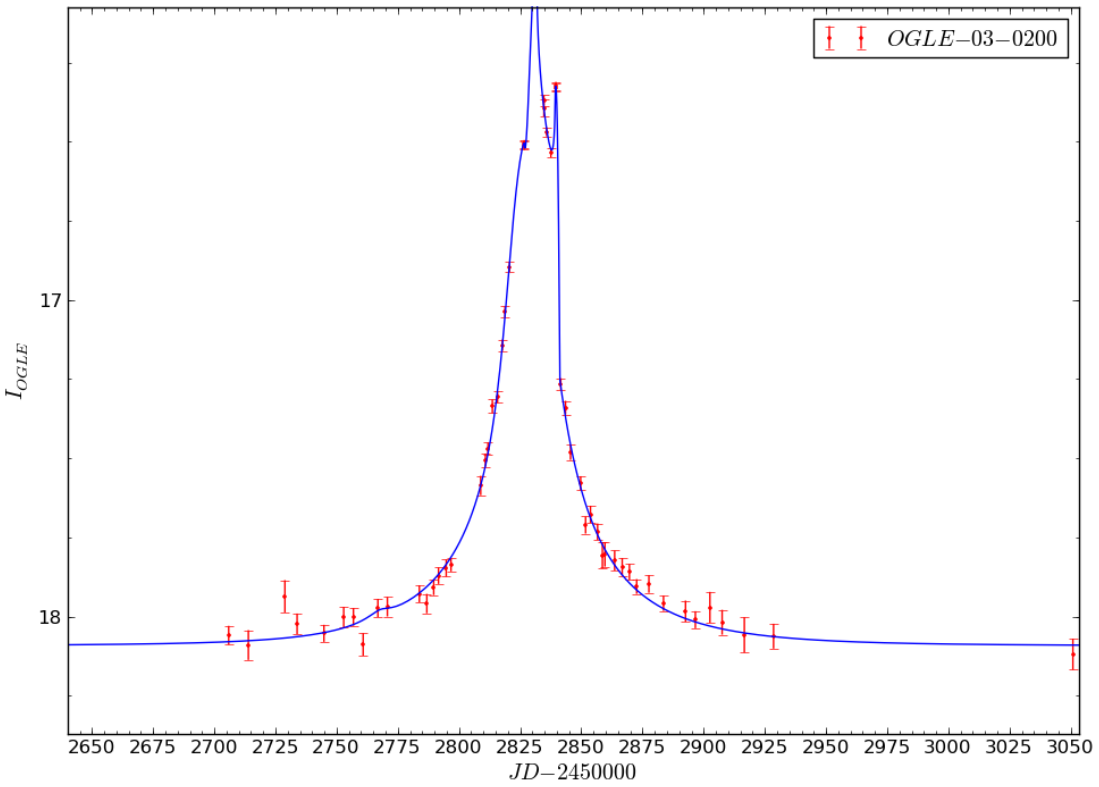


FIGURE 7.38: A light curve of the model solution (Table 7.9) of OGLE-2003-BLG-0200 determined by the GPU search method.

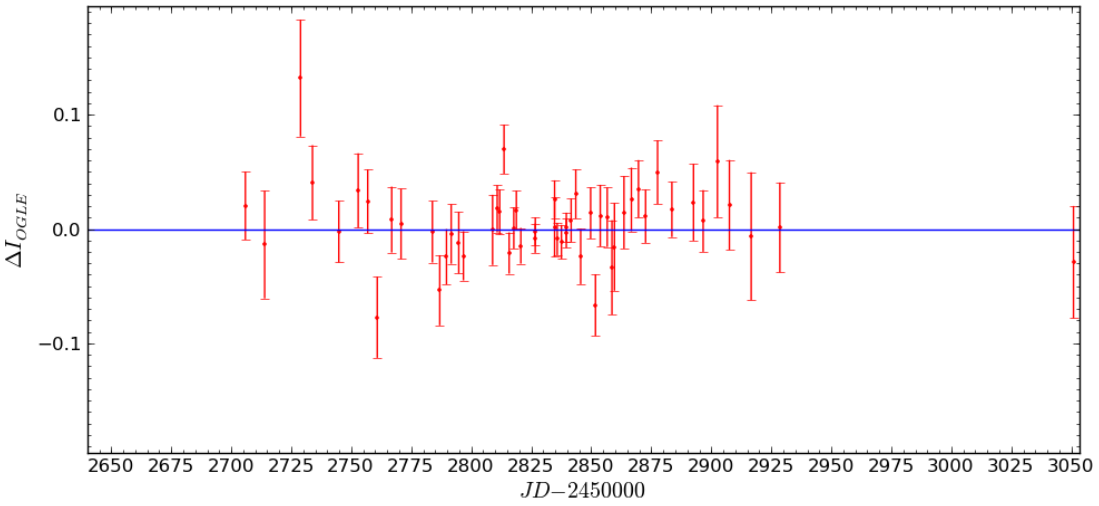


FIGURE 7.39: Light curve residuals of the model solution (Table 7.9) of OGLE-2003-BLG-0200 determined by the GPU search method.

The solution set proposed by M. Jaroszyński (shown in Table 7.9) is the wide degenerate pair of the close solution found by the GPU search method. A comparable close solution was explored by the local MCMC chains and was disfavored for the wide solution by a $\Delta\chi^2$ of 1.28. This minimal difference shows there is not enough data to favour either the close or wide solution for this event.

7.2.10 OGLE-2003-BLG-0236

The modelling routine of this event was started by providing 46 data points between the AJDs of 2100.0 and 2550.0, and the requirements that the model must contain at least two peaks located near 2400.79919 and 2416.9, and achieve a minimum magnitude change of at least 1.0. The analysis of this event took 2 hours and 17 minutes to perform a d, q grid search and MCMC searchs (Figure 7.40) in areas of local minima. This resulted in the parameter set detailed in Table 7.10, were the errors are determined from an EMCEE search of an error-normalized data set. The binary lens solution produces a caustic structure, light curve, and residuals shown in Figures 7.41-7.43.

TABLE 7.10: Binary lens model solutions of OGLE-2003-BLG-0236, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.838	0.8386 ± 0.0043	0.804 ± 0.034
q	0.175	0.1750 ± 0.0031	0.131 ± 0.028
ρ	–	1.1×10^{-7}	$8.0 \pm 3.4 \times 10^{-6}$
u_0	-0.13	-0.1281 ± 0.0016	0.103 ± 0.014
ϕ	3.4271	3.4278 ± 0.0013	2.87 ± 0.072
t_0	2801.5	2801.503 ± 0.040	2801.0 ± 1.8
t_E	73.8	73.828 ± 0.050	88.3 ± 9.6
χ^2	473.825	471.655	474.388

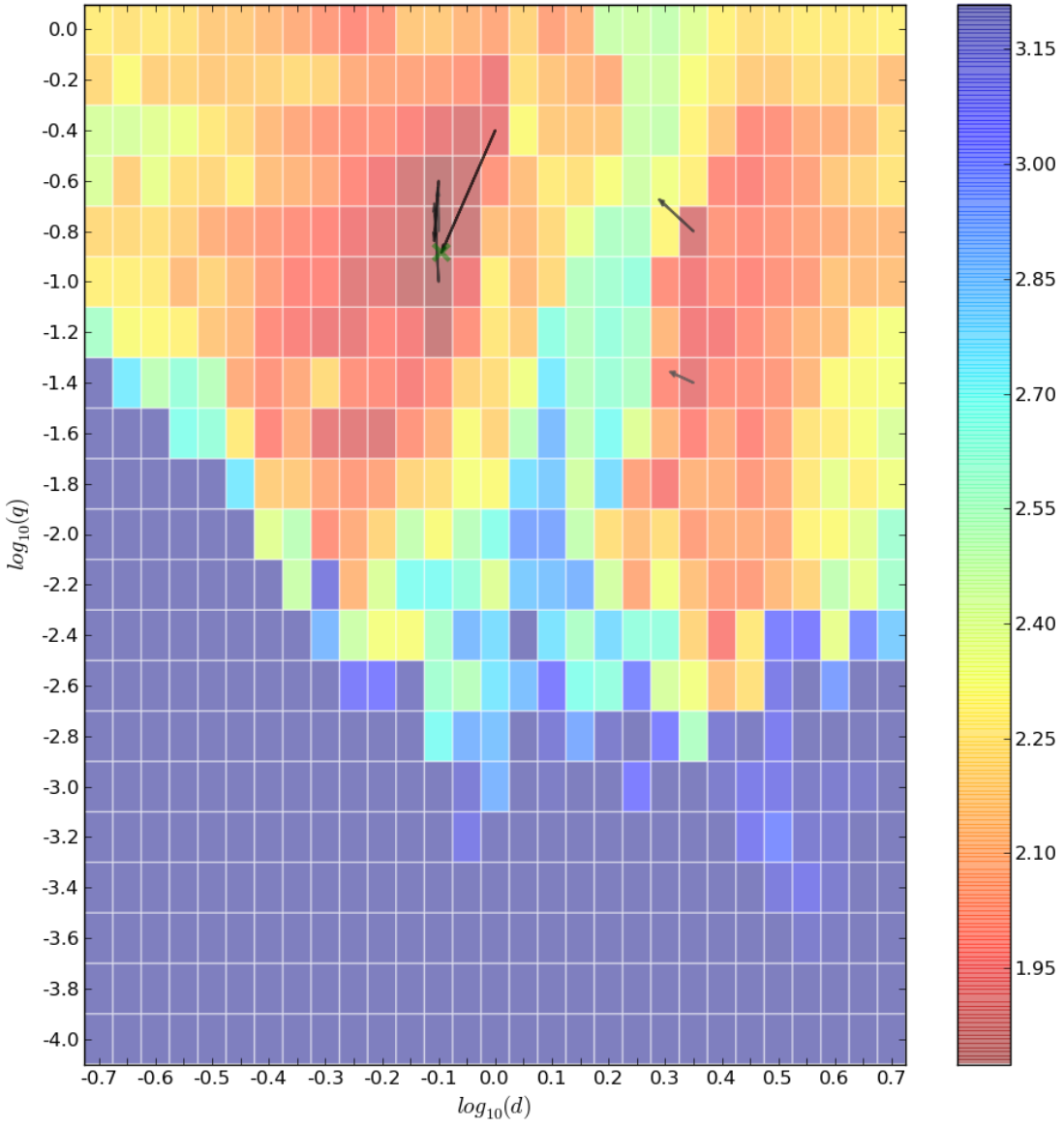


FIGURE 7.40: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0236.

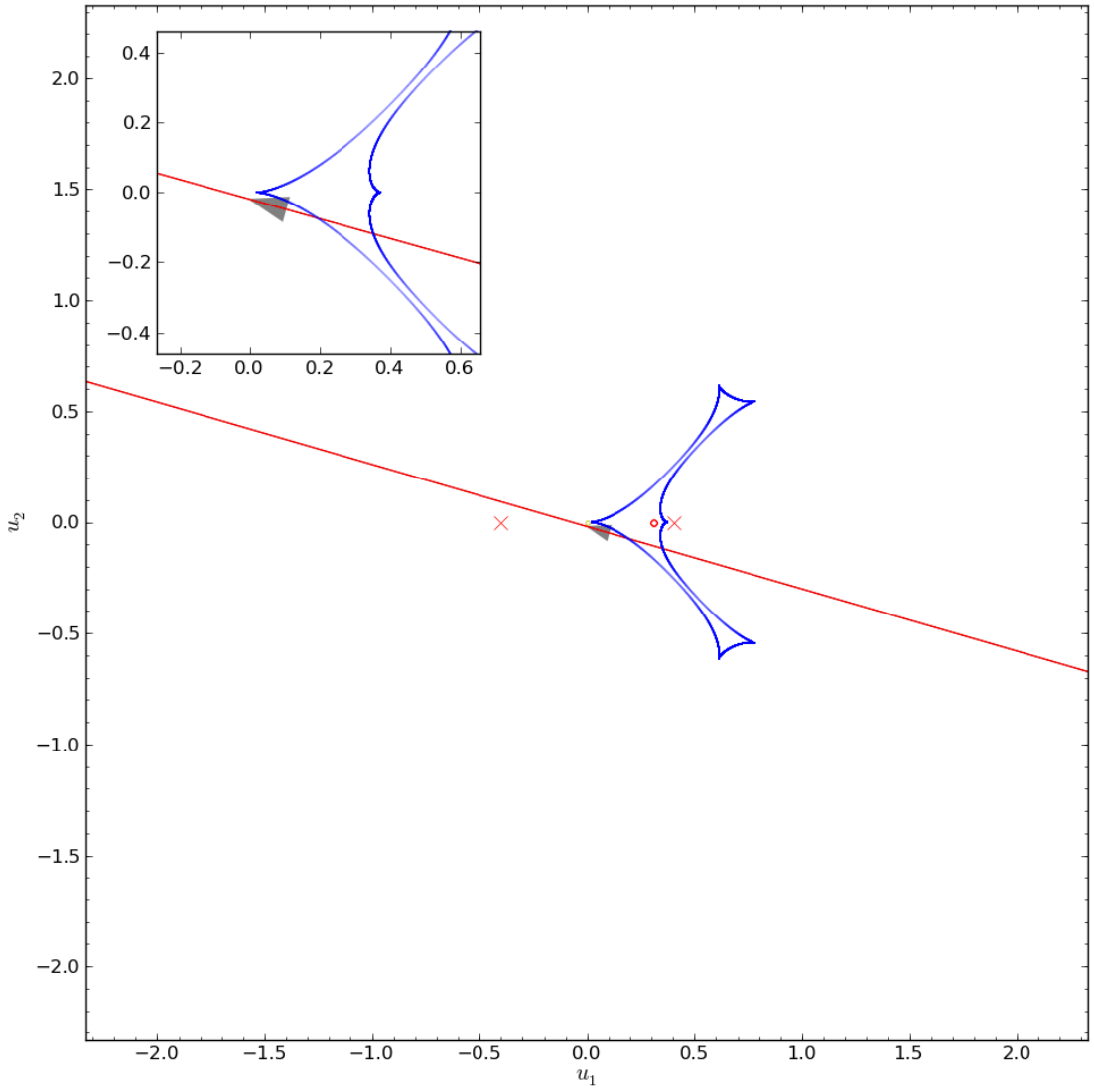


FIGURE 7.41: A caustic map with overlaid trajectory for the model solution (Table 7.10) of OGLE-2003-BLG-0236 determined by the GPU search method.

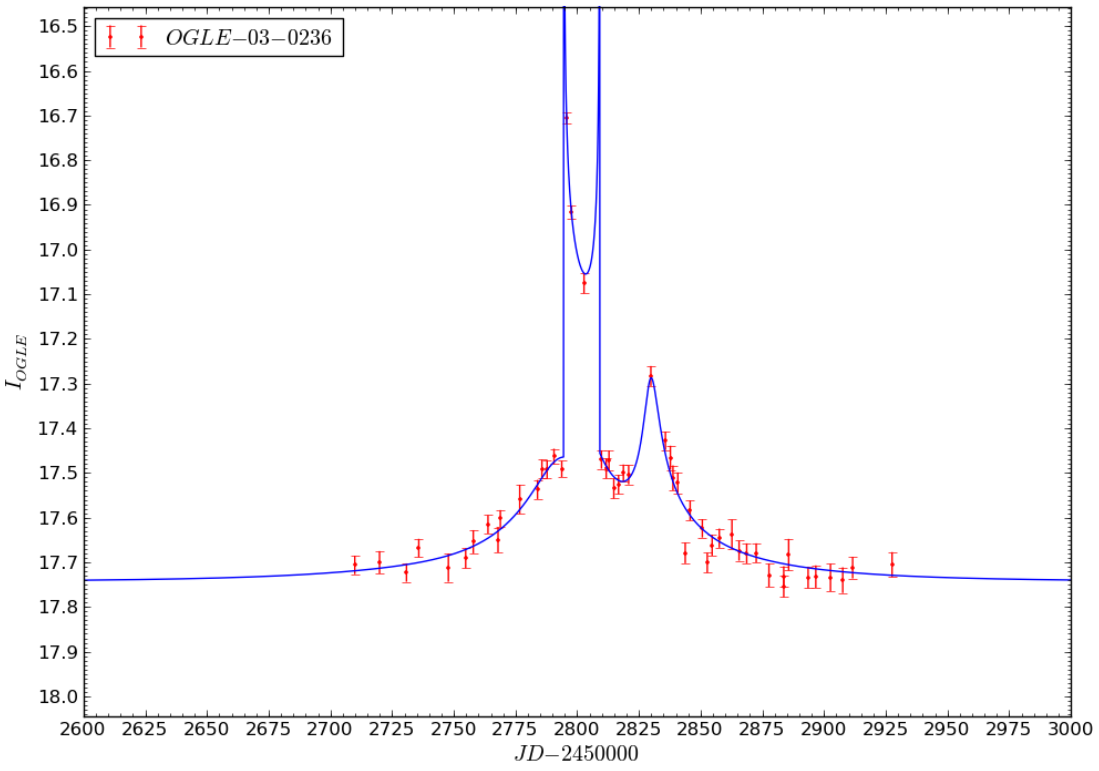


FIGURE 7.42: A light curve of the model solution (Table 7.10) of OGLE-2003-BLG-0236 determined by the GPU search method.

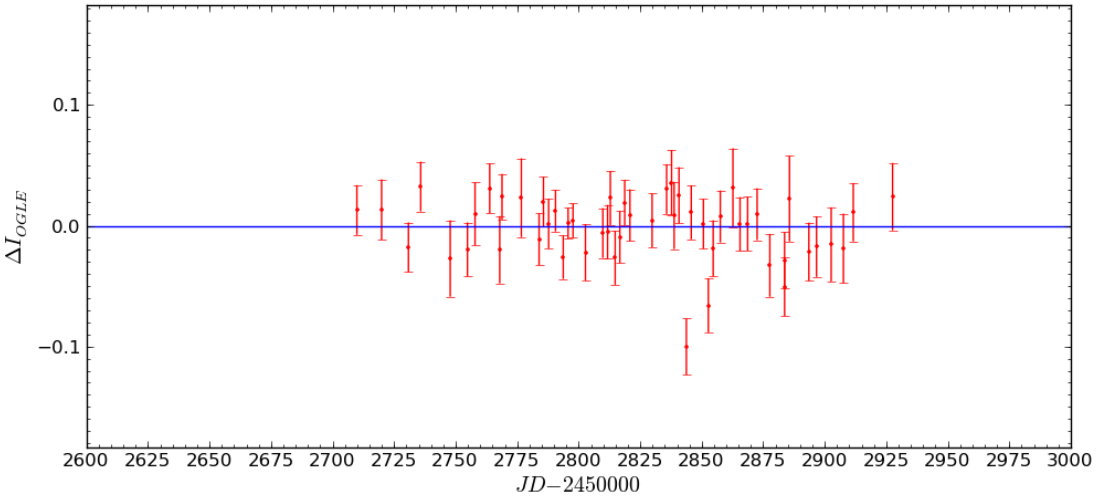


FIGURE 7.43: Light curve residuals of the model solution (Table 7.10) of OGLE-2003-BLG-0236 determined by the GPU search method.

Table 7.9 shows the parameter set presented by M. Jaroszyński, which shows an agreement with the global minima identified by the GPU search method.

7.2.11 OGLE-2003-BLG-0260

To start the analysis of this event the modeler was provided with the requirements that a valid model must contain at least two peaks with one of them being located around the AJD of 2786.78185, and the model must achieve at least a minimum change in magnitude of 3.4. Using a total of 273 data points between the AJDs of 2000.0 and 3500.0, the search of all parameter space (Figure 7.44) was performed in 1 hour and 56 minutes. This found a global minimum with the parameter set shown in Table 7.11, which produce the caustic, light curve, and residuals Figures of 7.45-7.47. To determine the error terms for each parameter, an additional EMCEE search was performed around this parameter set using error-normalized data.

TABLE 7.11: Seven parameter binary lens model solution of OGLE-2003-BLG-0260, determined using the GPU modelling method detailed in this thesis.

Parameter	GPU solution
d	2.385 ± 0.066
q	0.348 ± 0.013
ρ	0.001005 ± 0.000044
u_0	1.308 ± 0.044
ϕ	1.196 ± 0.026
t_0	2938.0 ± 6.6
t_E	209.6 ± 7.0
χ^2	609.520

TABLE 7.12: M. Jaroszyński's binary lens point source six parameter model solution of OGLE-2003-BLG-0260.

Parameter	Solution 1	Solution 2	Solution 3
d	2.269	1.845	0.624
q	0.112	0.488	0.450
u_0	-1.52	-0.42	-0.01
ϕ	5.0349	1.8635	4.4838
t_0	2968.7	2840.7	2827.8
t_E	272.3	139.4	106.7
χ^2	2115.332	2201.254	4282.890

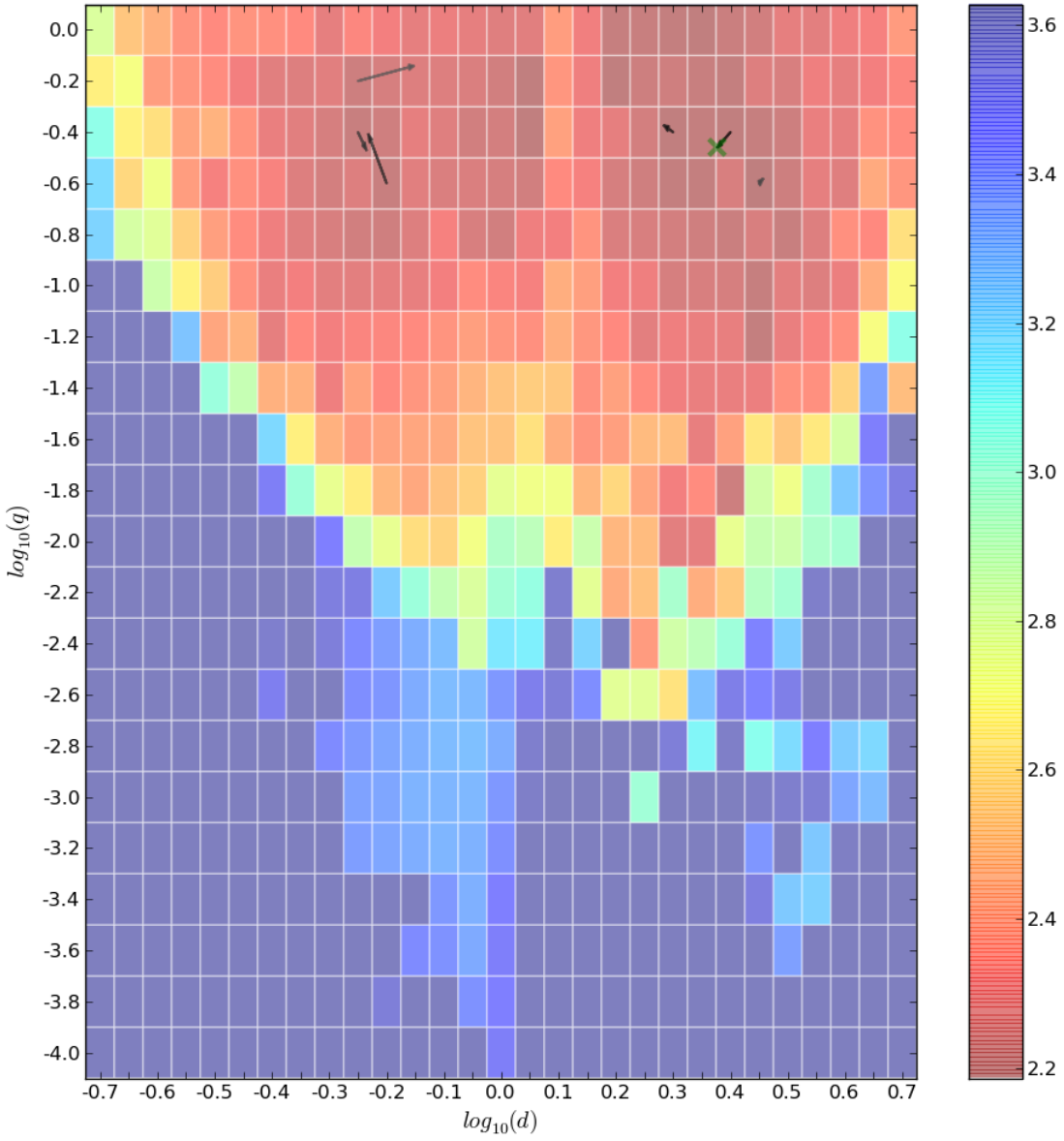


FIGURE 7.44: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0260.

TABLE 7.13: M. Jaroszyński's binary lens point source six parameter solutions of OGLE-2003-BLG-0260 after a minimization search performed with EMCEE.

Parameter	Solution 1 minimized	Solution 2 minimized	Solution 3 minimized
d	2.2713 ± 0.0029	1.8464 ± 0.0022	0.6236 ± 0.0013
q	0.1136 ± 0.0012	0.4891 ± 0.0025	0.4487 ± 0.0016
u_0	-1.52131 ± 0.0018	-0.4167 ± 0.0021	-0.0083 ± 0.0010
ϕ	5.03501 ± 0.00034	1.86330 ± 0.00038	4.48400 ± 0.00026
t_0	2968.685 ± 0.018	2840.699 ± 0.023	2827.799 ± 0.014
t_E	272.329 ± 0.038	139.397 ± 0.024	106.687 ± 0.013
χ^2	609.256	612.063	608.344

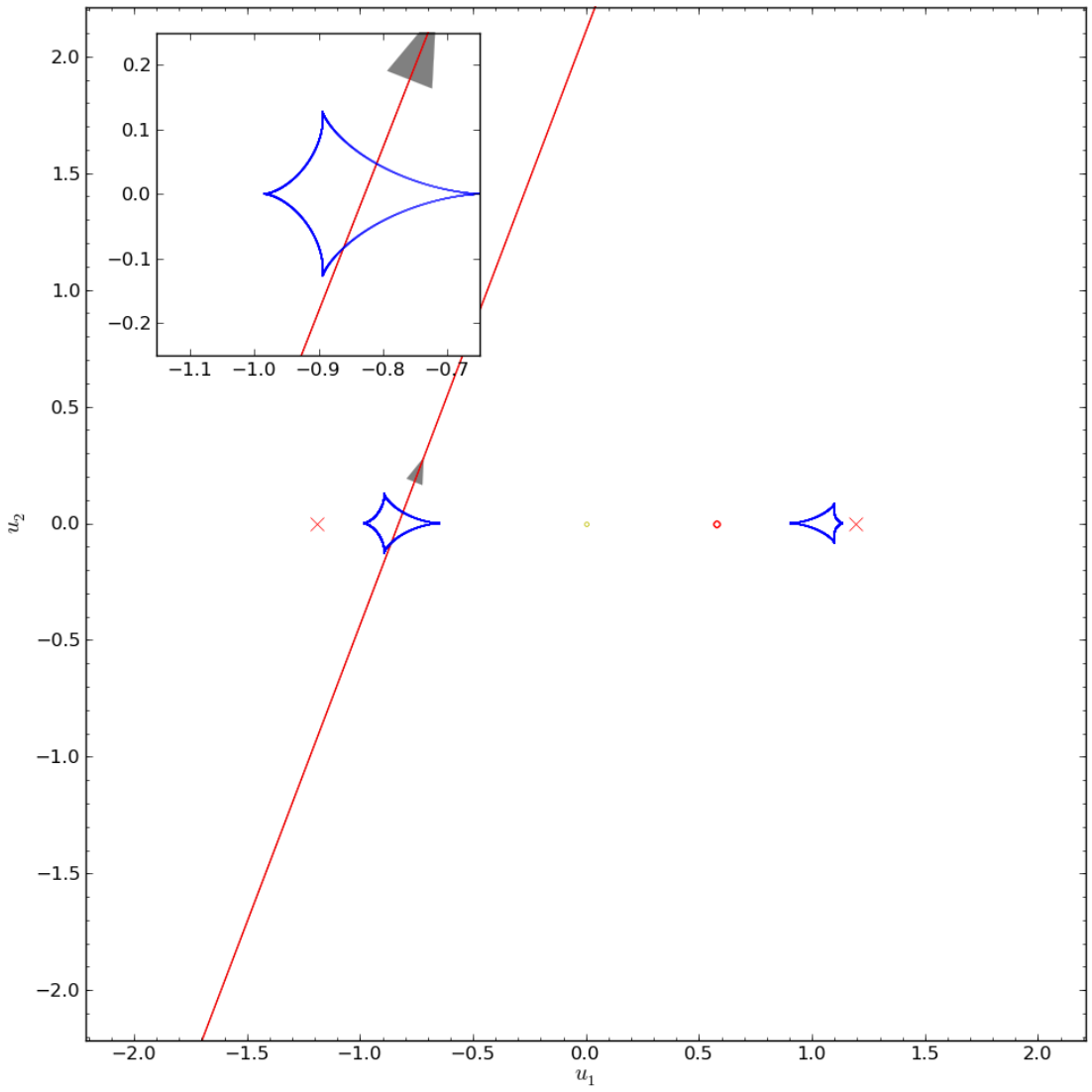


FIGURE 7.45: A caustic map with overlaid trajectory for the model solution (Table 7.11) of OGLE-2003-BLG-0260 determined by the GPU search method.

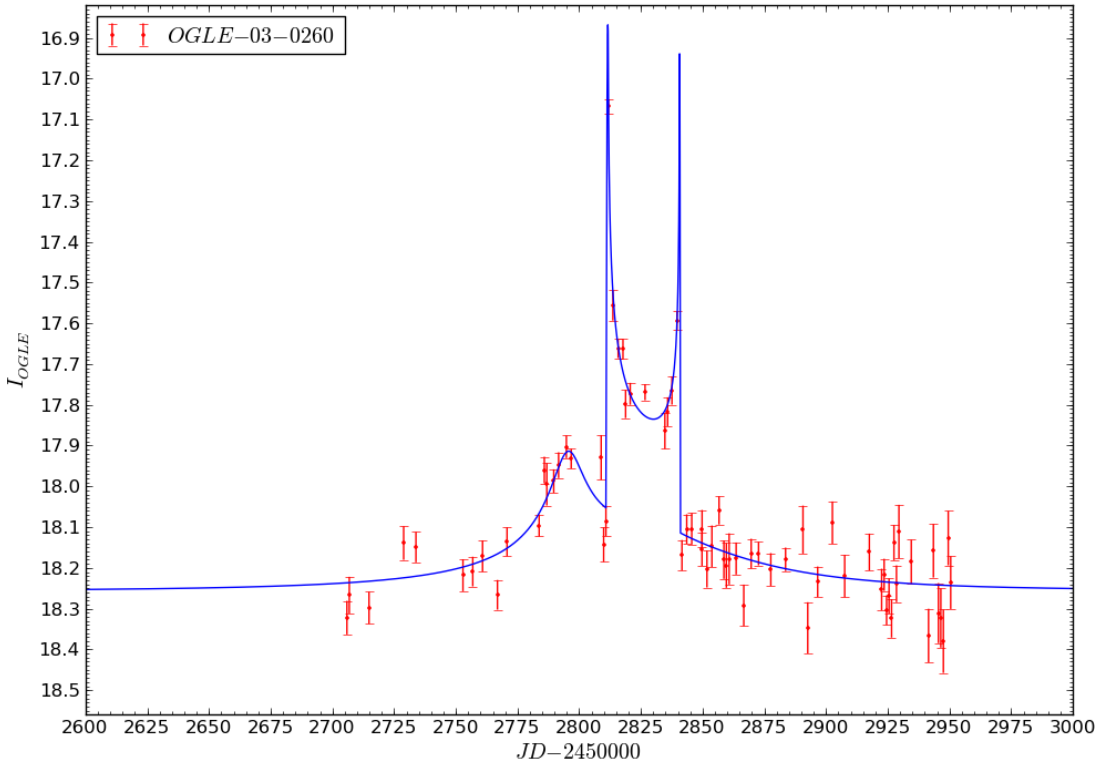


FIGURE 7.46: A light curve of the model solution (Table 7.11) of OGLE-2003-BLG-0260 determined by the GPU search method.

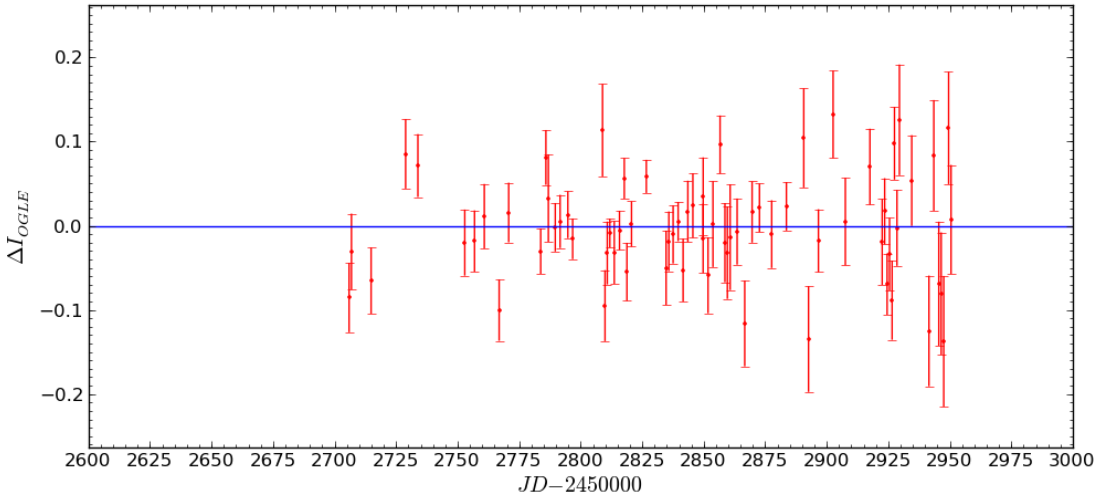


FIGURE 7.47: Light curve residuals of the model solution (Table 7.11) of OGLE-2003-BLG-0260 determined by the GPU search method.

M. Jaroszyński presented three different solutions (shown in Table 7.12), stating that all had similar χ^2 values. In comparison with the GPU search, M. Jaroszyński's Solution 1 was most similar to the parameter set found, with both of the other solutions identified as local minima and explored using MCMC, shown as chains 2 and 3 in Table 7.14 and Figure 7.48. All three parameter sets produced comparable χ^2 values, with a $\Delta\chi^2$ difference of only 5.5 between all three. This small $\Delta\chi^2$ range shows that the parameter space is not well constrained and several solutions are valid.

TABLE 7.14: Seven parameter binary lens local minima model solutions of OGLE-2003-BLG-0260, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	2.38471	1.92453	0.58650	0.58161	2.86103	0.70264
q	0.34767	0.42202	0.38968	0.33935	0.26095	0.72199
ρ	0.00100	1.10×10^{-6}	0.00160	0.00086	0.00012	$1.47\times^{-7}$
u_0	1.30829	−0.40455	−0.00291	−0.00555	1.84304	−0.04575
ϕ	1.19640	1.86204	4.48282	4.44908	1.22508	4.51447
t_0	2937.963	2843.129	2826.865	2826.418	3045.620	2829.695
t_E	209.621	163.494	134.147	145.014	324.213	84.873
χ^2	609.520	611.922	615.024	615.715	617.572	1823.450

7.2.12 OGLE-2003-BLG-0267

For a valid model to be selected in this parameter search, the solution is required to have at least fours peaks, where two of them are located close to AJDs 2848.52 and 2852.0, and achieve a minimum change in magnitude of at least 3.75. Data was provided between the AJDs of 2000.0 and 3500.0, which contained 256 points. The search (Figure 7.49) of parameter space took 5 hours and 19 minutes locating the solution shown in Table 7.15, and Figures 7.50-7.52. This was followed by an EMCEE search of error-normalized data to determine the error terms for each parameter. Due to the strict requirements imposed on the model, the initial d, q grid search was performed quickly in 34 minutes. However, the complex structure of the light curve with multiple peaks means the CICIRAS modelling method is slower than average and took longer than normal to complete the six local MCMC searchs.

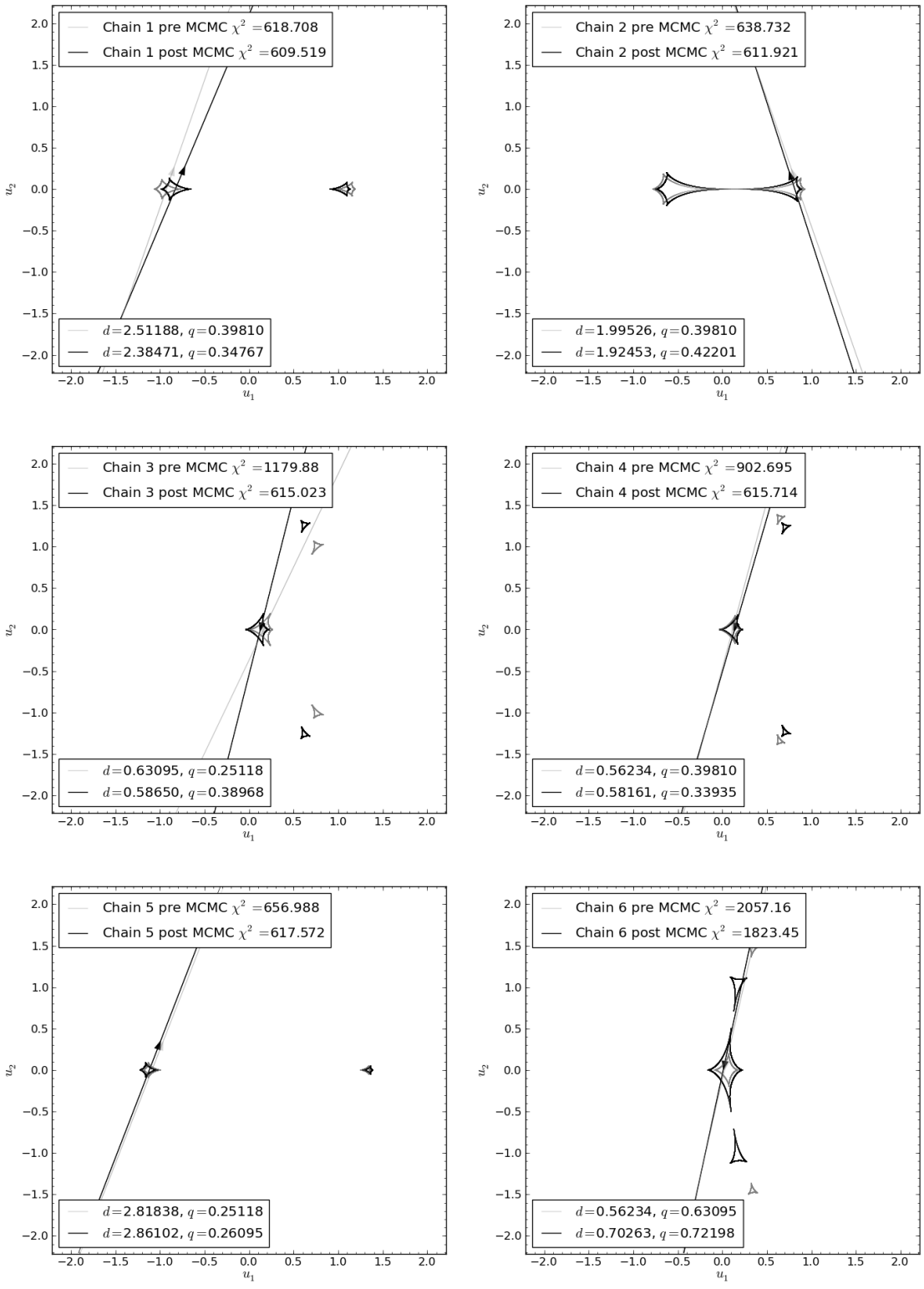


FIGURE 7.48: The caustic and trajectory of the solutions found (Table 7.14) after performing a local area MCMC search for the best six pixels from the d, q grid search of OGLE-2003-BLG-0260.

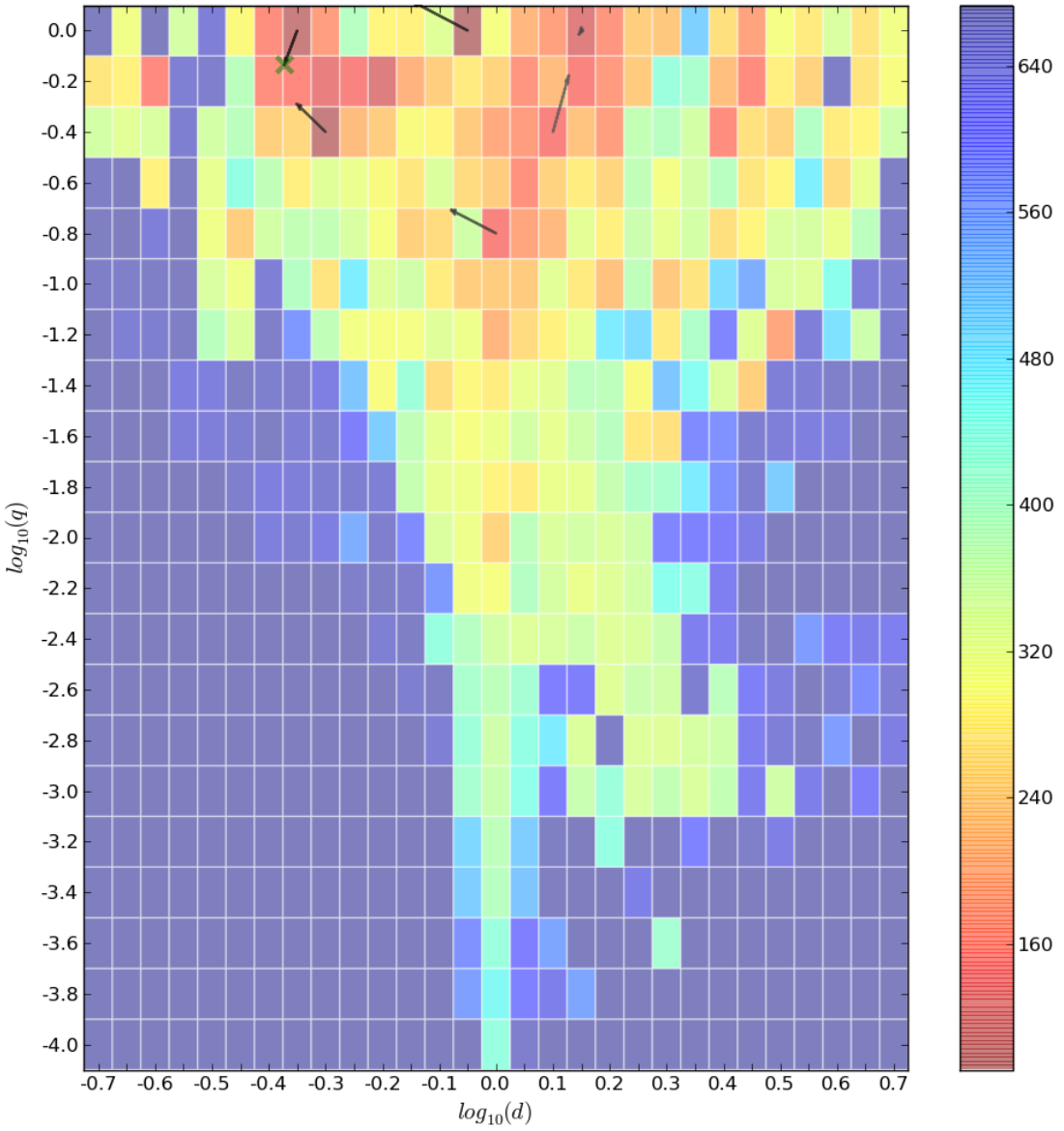


FIGURE 7.49: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0267.

TABLE 7.15: Binary lens model solutions of OGLE-2003-BLG-0267, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.352	0.35096 ± 0.00050	0.424 ± 0.020
q	0.628	0.62647 ± 0.00050	0.732 ± 0.020
ρ	0.0008	0.00076 ± 0.00042	0.001217 ± 0.000054
u_0	-0.01	-0.00826 ± 0.00027	0.01167 ± 0.00039
ϕ	1.5338	1.53411 ± 0.00011	4.74 ± 0.18
t_0	2845.3	2845.3327 ± 0.0059	2845.3 ± 1.3
t_E	88.8	88.8113 ± 0.0072	56.3 ± 1.7
χ^2	188699.953	15451.226	736.260

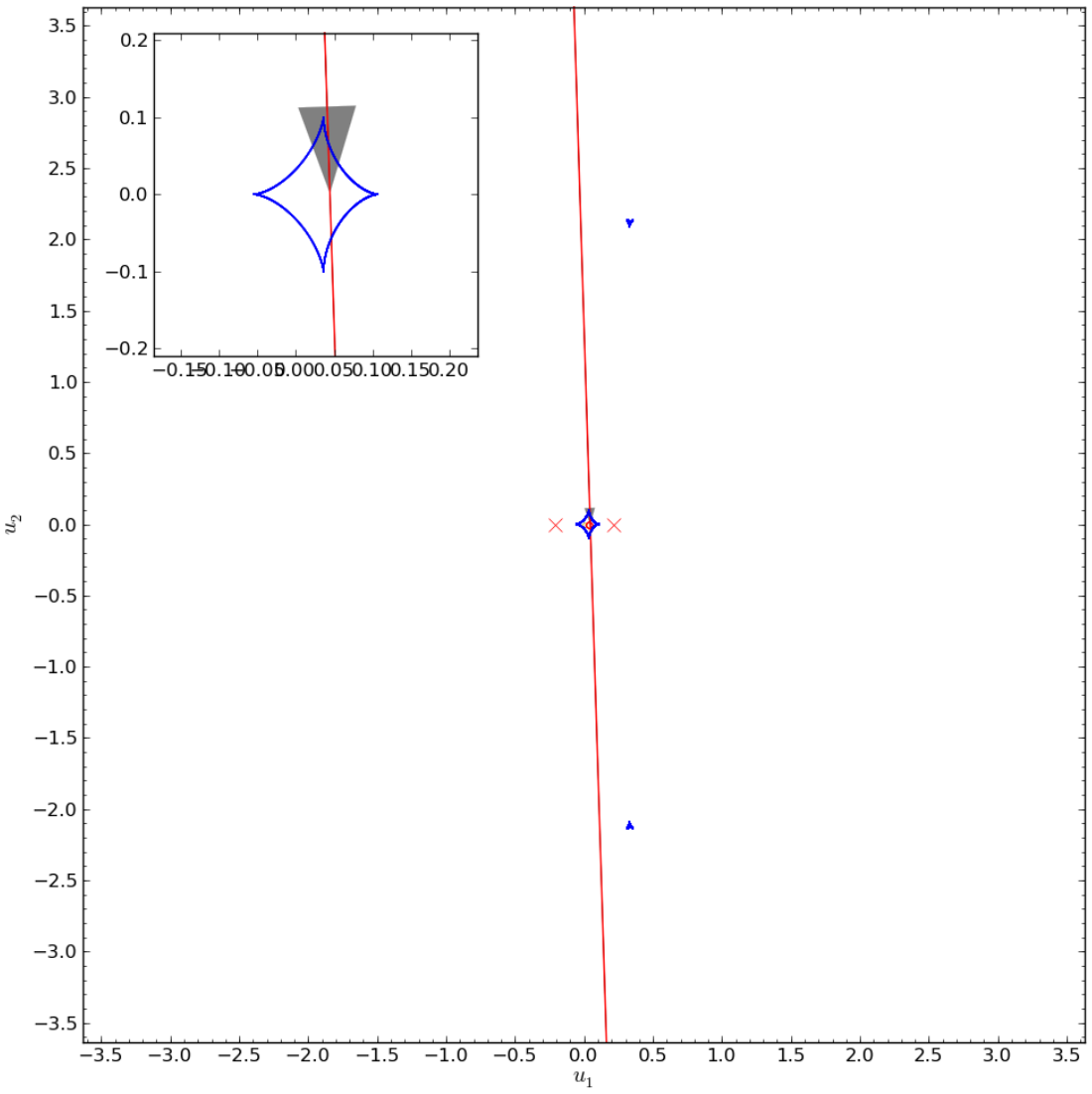


FIGURE 7.50: A caustic map with overlaid trajectory for the model solution (Table 7.15) of OGLE-2003-BLG-0267 determined by the GPU search method.

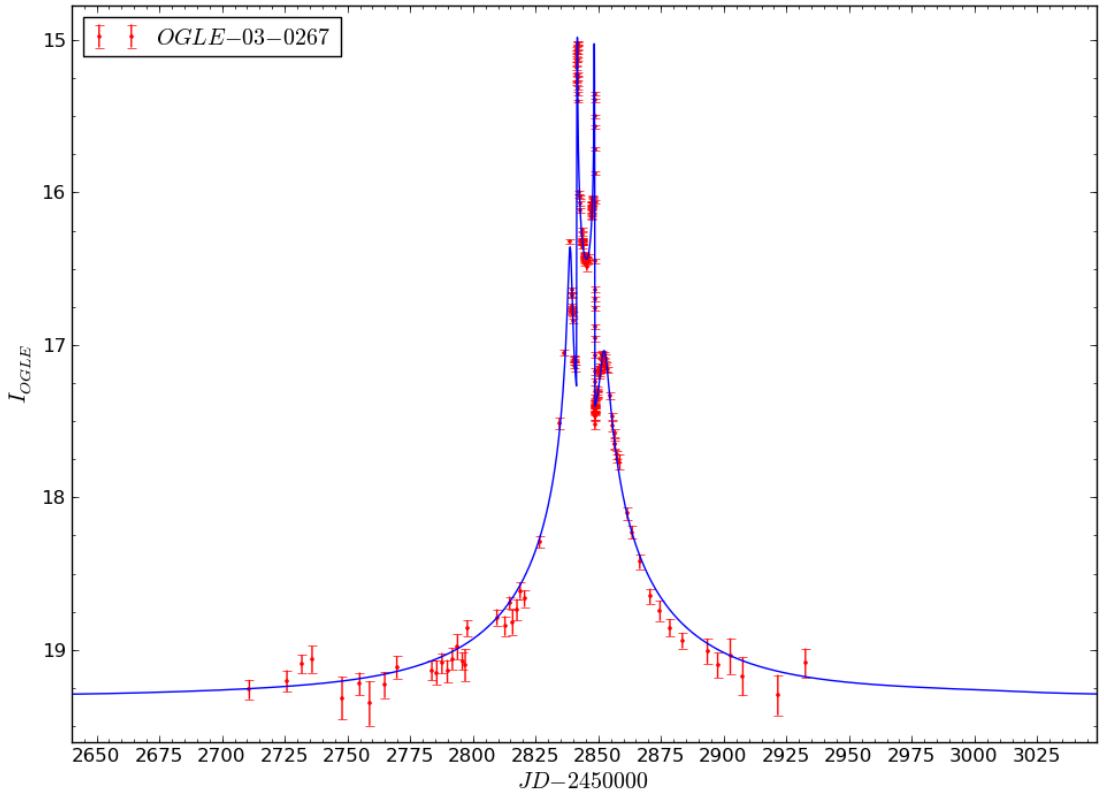


FIGURE 7.51: A light curve of the model solution (Table 7.15) of OGLE-2003-BLG-0267 determined by the GPU search method.

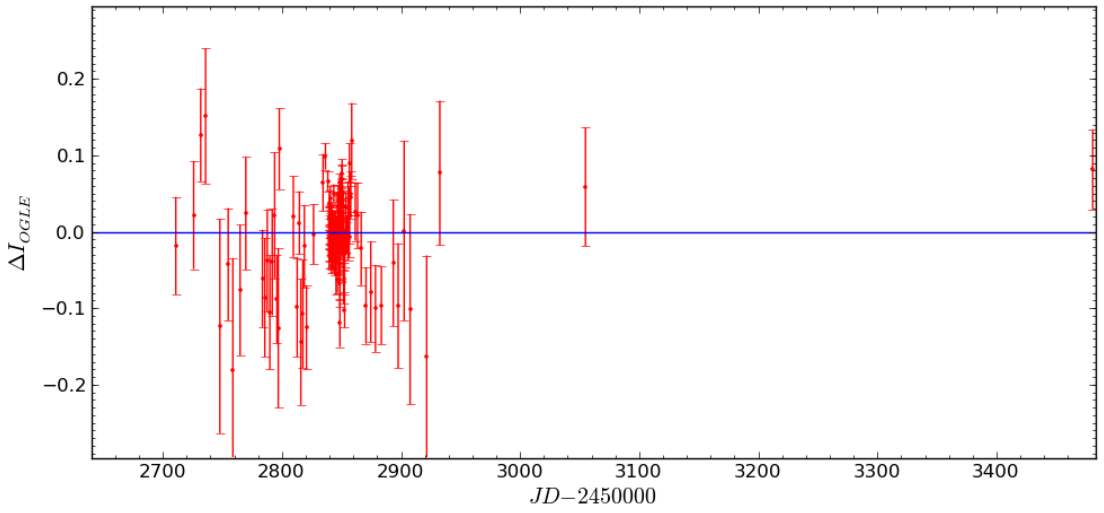


FIGURE 7.52: Light curve residuals of the model solution (Table 7.15) of OGLE-2003-BLG-0267 determined by the GPU search method.

Table 7.15 shows the best fitting parameter set presented by M. Jaroszyński, which is close to the one found by the GPU. The solutions are symmetrical across the x-axis, which can be mirrored by taking the trajectory angle as $2\pi - \phi$ and closest approach as $-u_0$.

7.2.13 OGLE-2003-BLG-0291

The parameter search of this event fitted a model to 208 data points between the AJDs of 2400 and 3200. The requirements for a valid solution are to achieve a minimum change in magnitude of at least 3.4 and to contain two or more peaks, with two being located around 3098.74207 and 3100.9154. The search (Figure 7.53) took 1 hour and 47 minutes to complete and identified the parameter set shown in Table 7.16, which produces Figures 7.54-7.56. An EMCEE search was carried out using an error-normalized set of data to identify the errors of each parameter.

TABLE 7.16: Binary lens model solutions of OGLE-2003-BLG-0291, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	3.457	3.177 ± 0.015	5.07 ± 0.25
q	0.837	1.103 ± 0.015	0.980 ± 0.045
ρ	–	1.1×10^{-7}	$6.75 \pm 0.35 \times 10^{-5}$
u_0	0.48	0.2791 ± 0.0079	-0.0997 ± 0.0059
ϕ	3.4650	3.4990 ± 0.0013	0.0370 ± 0.0016
t_0	2956.6	2955.156 ± 0.056	2972.6 ± 2.0
t_E	39.8	39.687 ± 0.075	52.6 ± 2.8
χ^2	177919.147	177827.887	2235.729

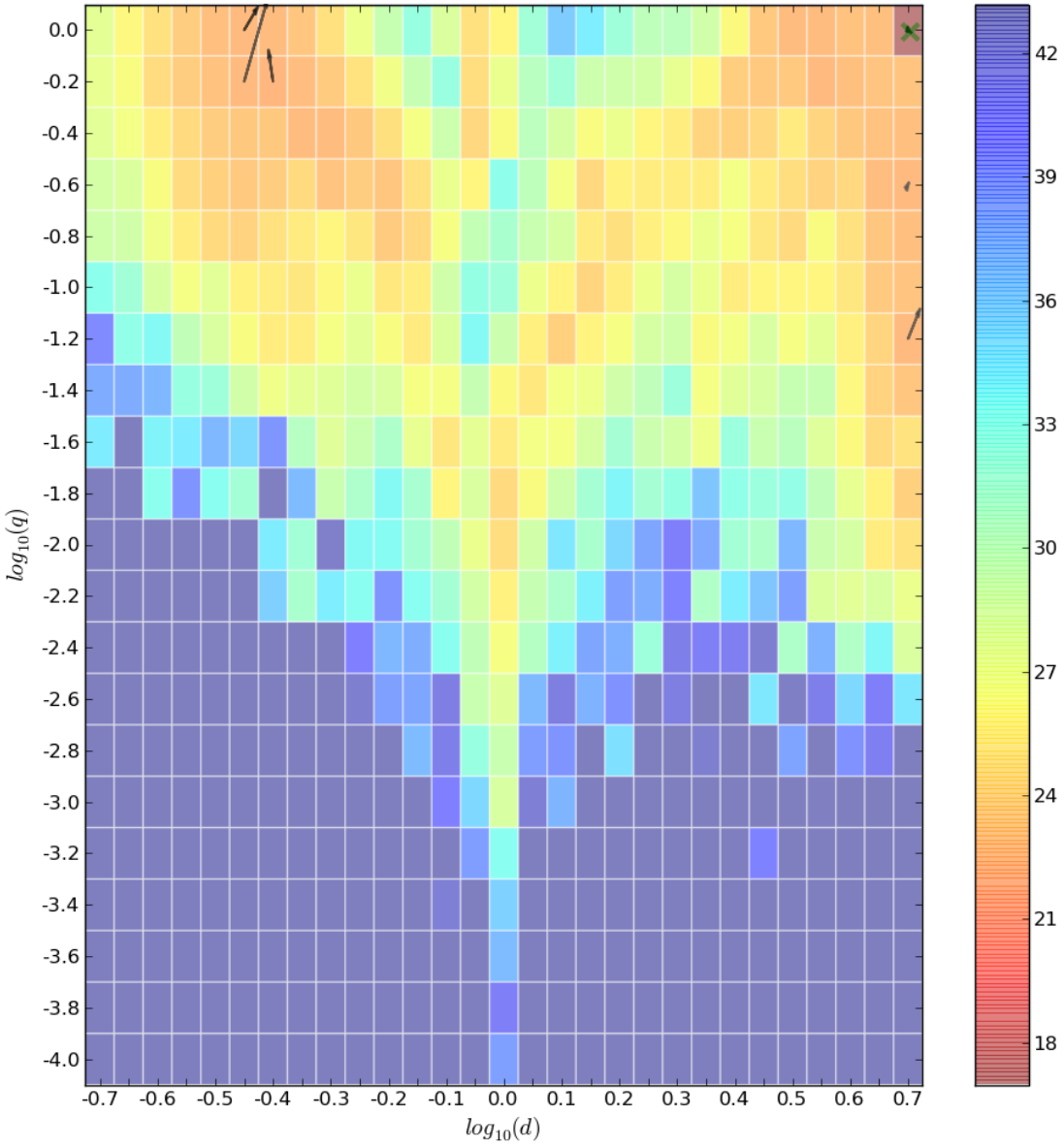


FIGURE 7.53: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0291.

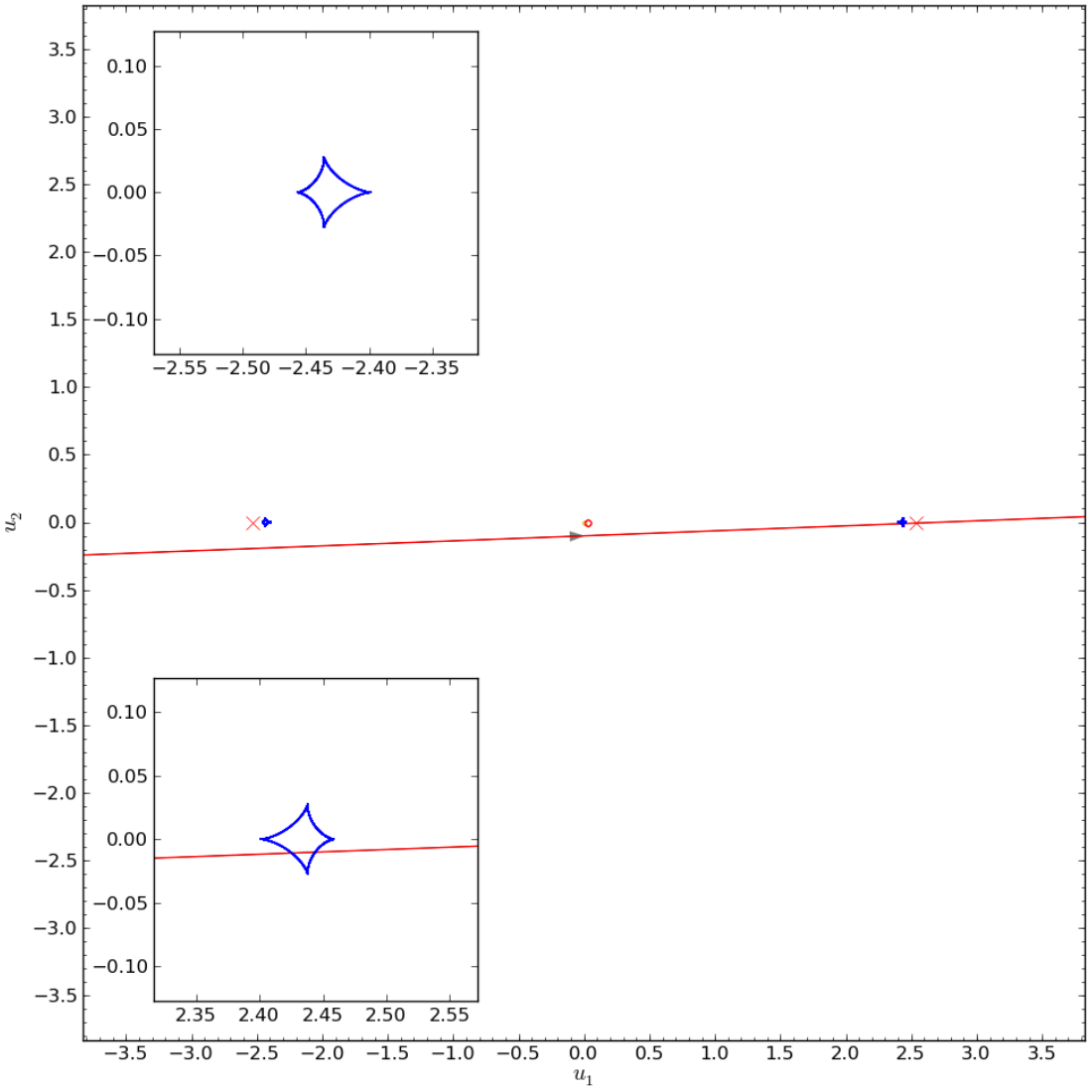


FIGURE 7.54: A caustic map with overlaid trajectory for the model solution (Table 7.16) of OGLE-2003-BLG-0291 determined by the GPU search method.

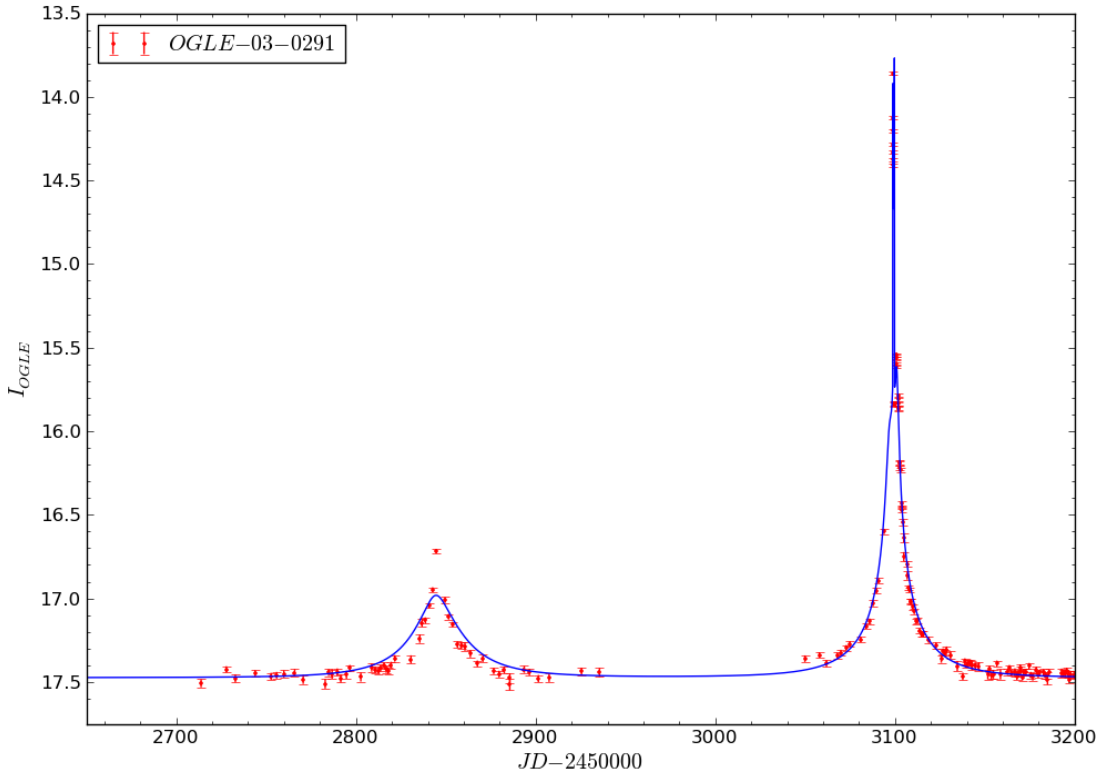


FIGURE 7.55: A light curve of the model solution (Table 7.16) of OGLE-2003-BLG-0291 determined by the GPU search method.

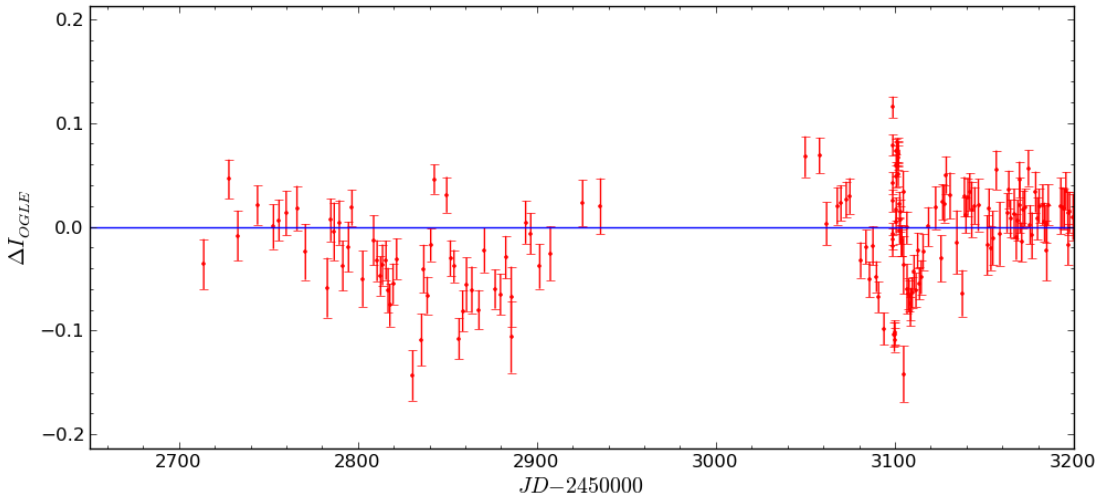


FIGURE 7.56: Light curve residuals of the model solution (Table 7.16) of OGLE-2003-BLG-0291 determined by the GPU search method.

M. Jaroszyński's model is different to the one determined by the GPU search technique. Upon comparison, a reproduction of the light curve proposed by M. Jaroszyński has not been possible, and despite an EMCEE minimization search, no comparable solution has been found. Visual inspection of the trajectory and light curve presented by Jaroszyński et al. (2004), suggests that it is unlikely the parameter set proposed correctly modeled the peak in the previous year located at AJD 2844. As can be seen in the GPU solution presented (Figure 7.55), the light curve does not fit to the data as well as would be expected, together with the long timescale of this event (over 400 days), this suggests that the parallax and/or lens-rotation are present.

7.2.14 OGLE-2003-BLG-0380

Modelling of this event was started with the requirement that the solution contains at least three peaks, with two being close to the AJDs of 2871.7 and 2879.7, and the model needs to achieve at least a 0.75 change in magnitude. Model fitting was performed on 185 data points between the AJDs 2400.0 and 3200.0. A grid search and six local MCMC explorations (Figure 7.57) took a total of 1 hour and 42 minutes to identify the global minimum, which was followed by an EMCEE search on error-normalized data to calculate the parameter's errors. The result of this search is shown in Table 7.17, which produces the caustic structure and trajectory, light curve, and residuals of Figures 7.58-7.60.

TABLE 7.17: Binary lens model solutions of OGLE-2003-BLG-0380, comparing M. Jaroszyński's original solution with an emcee minimized solution based around his original, with one determined using the GPU modelling method.

Parameter	Jaroszyński's	Jaroszyński's minimized	GPU
d	0.784	0.7814 ± 0.0073	0.736 ± 0.048
q	0.615	0.6115 ± 0.0055	0.50 ± 0.11
ρ	–	1.1×10^{-7}	0.0034 ± 0.0013
u_0	0.14	0.141 ± 0.010	-0.127 ± 0.021
ϕ	2.9505	2.9505 ± 0.0017	3.348 ± 0.053
t_0	2876.5	2876.50 ± 0.10	2876.2 ± 1.2
t_E	78.9	78.886 ± 0.071	87 ± 11
χ^2	3533.999	237.177	239.948

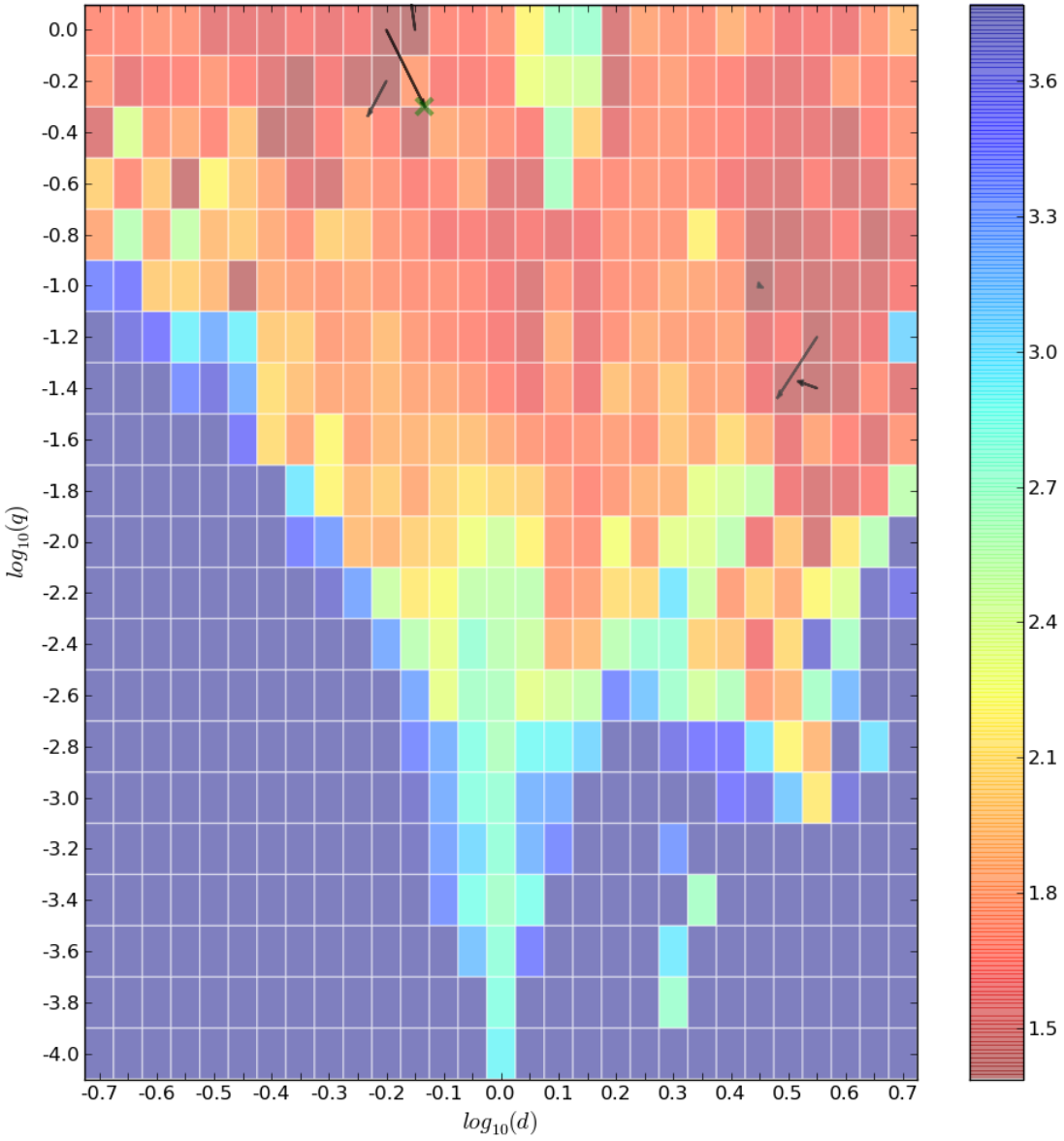


FIGURE 7.57: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0380.

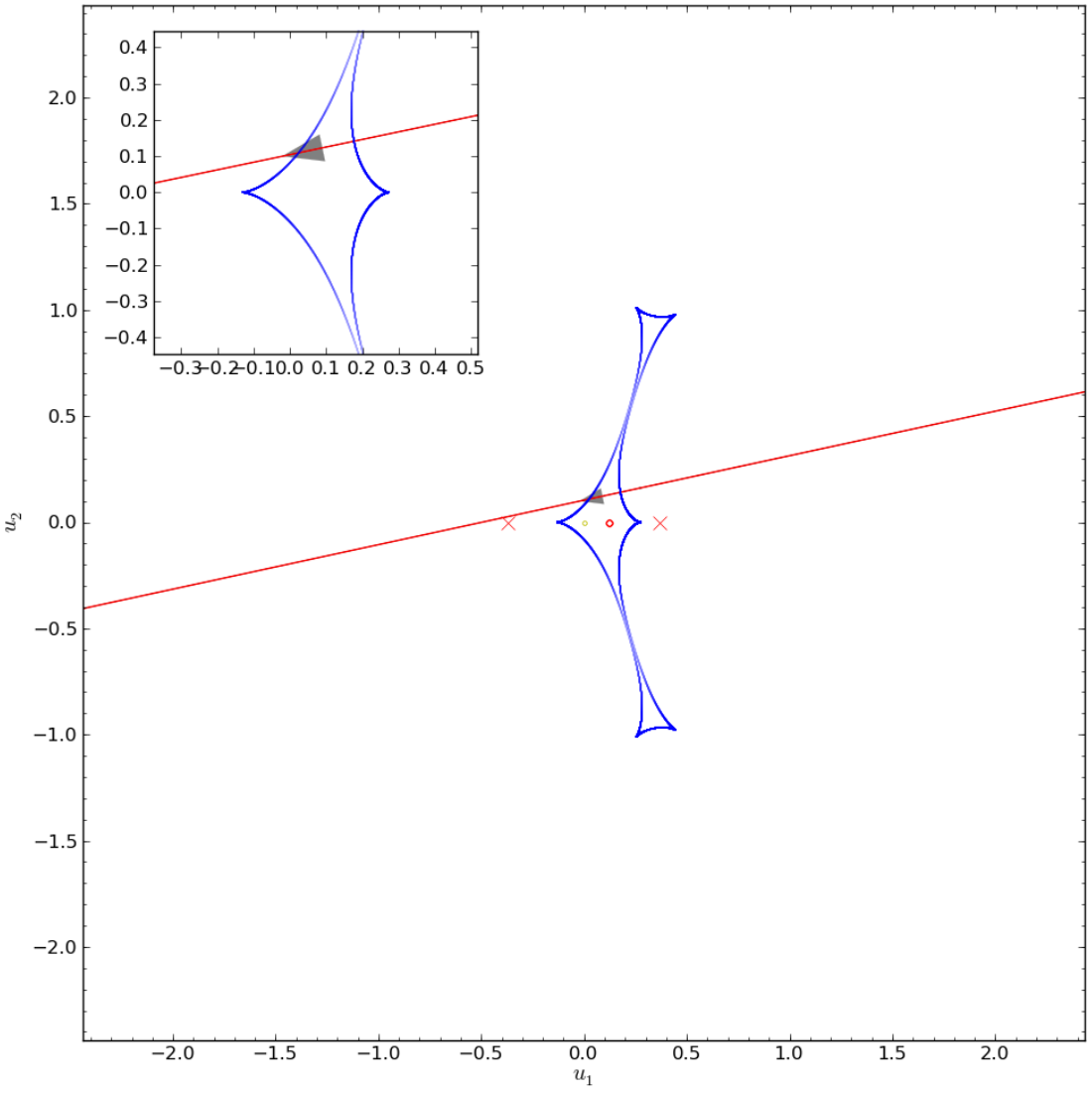


FIGURE 7.58: A caustic map with overlaid trajectory for the model solution (Table 7.17) of OGLE-2003-BLG-0380 determined by the GPU search method.

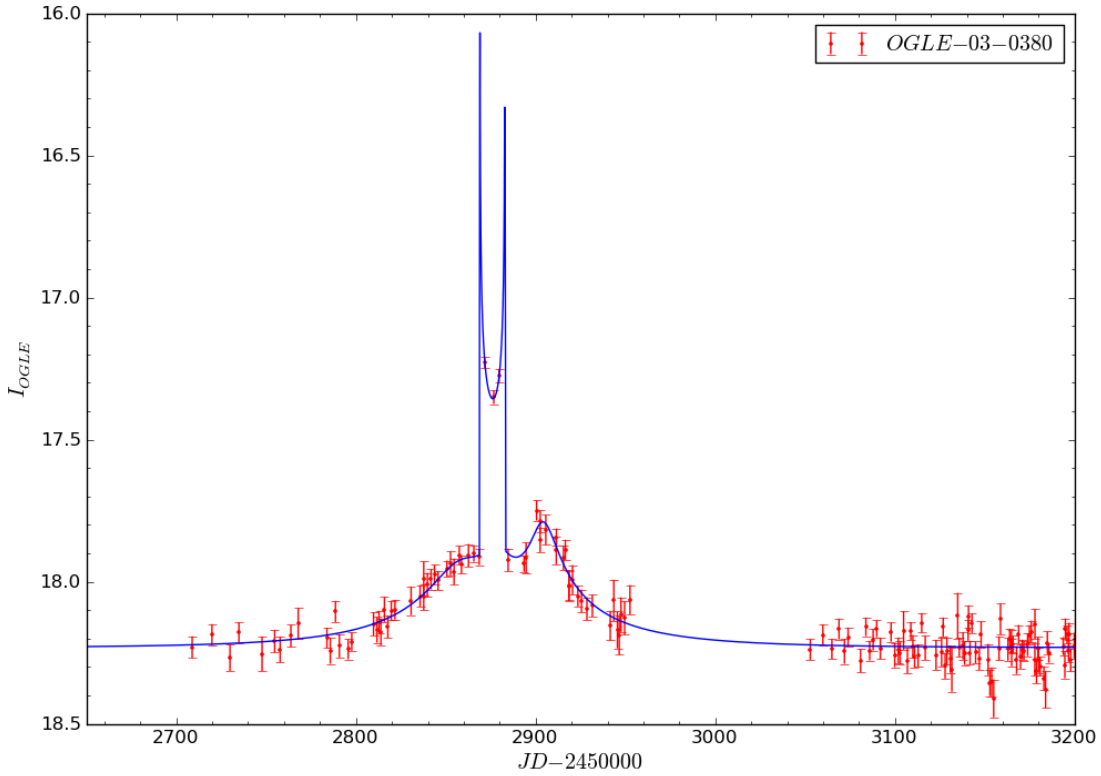


FIGURE 7.59: A light curve of the model solution (Table 7.17) of OGLE-2003-BLG-0380 determined by the GPU search method.

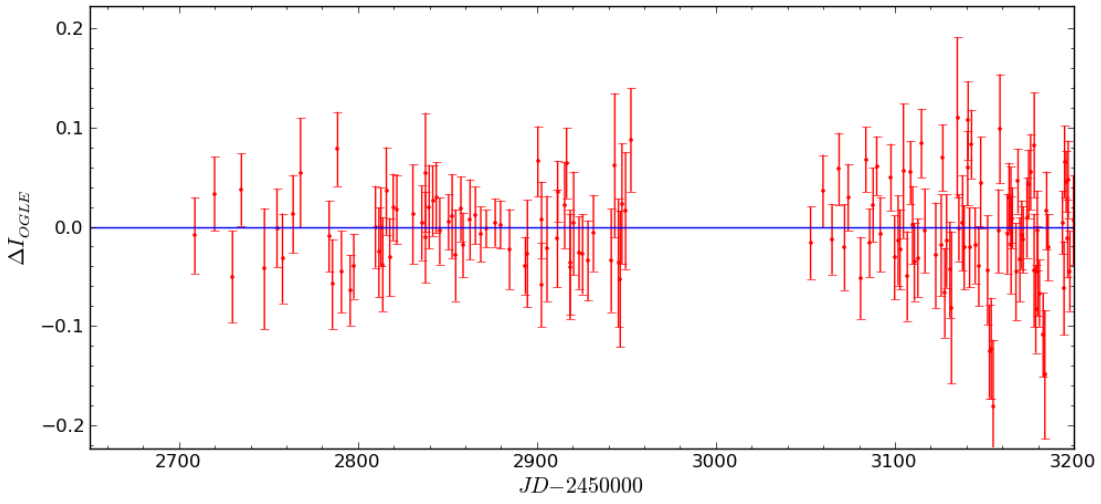


FIGURE 7.60: Light curve residuals of the model solution (Table 7.17) of OGLE-2003-BLG-0380 determined by the GPU search method.

Both the model found by the GPU search method and the one presented by M. Jaroszyński locate the same global minima, showing parameter sets which are similar, but mirrored across the symmetrical x-axis.

7.3 MOA-2003-BLG-0053/OGLE-2003-BLG-0235

OGLE first detected this event on 2003 June 22 and labeled it as OGLE-2003-BLG-0235. At that time it appeared to be a standard single lens model. However, a month later the data deviated from a single lens model, showing the characteristic signature of a caustic crossing, which alerted the MOA group. The caustic crossing lasted seven days and was observed by the survey groups with high cadence before it exited the caustic. Both survey groups continued to observe this event regularly as the light curve continued to follow the shape of a single lens model. The initial interpretation of the short timescale and small amplitude deviation of the caustic crossing suggested a large mass ratio binary system. Three independent modelling processes were carried out on this data to confirm the results, which were then presented in Bond et al. (2004). More recently this event has been modeled by Bennett (2010) using a previously unpublished modelling method. The new GPU search method is used here to show agreement between our new search method and previous modelling procedures.

All previous models agreed on a planetary solution shown in the first column of Table 7.18, with a close fitting alternate model and the best non-planet model being shown in the second and third columns respectively. These binary lens parameter sets represent the caustic structure and trajectories that produce the caustic trajectories, light curves, and residuals seen in Figures 7.61 - 7.69.

TABLE 7.18: Binary lens seven parameter model solutions of OGLE-2003-BLG-0235, showing planetary solutions and best non-planetary model (Bond et al., 2004). χ^2 values determined by our model with $\Gamma_I = 0.53$ on survey telescope's alert page data between AJD 2300-3400.

Parameter	Planetary	Alternate Planet	Best non-planet
d	1.120(7)	1.121	1.090
q	0.0039(+11, −7)	0.0070	0.0300
ρ	0.00096(11)	0.00104	0.00088
u_0	0.133(3)	0.140	0.144
ϕ	0.7644(0.024)	0.6789	0.1379
t_0	2848.06(13)	2847.90	2846.20
t_E	61.5(1.8)	58.5	57.5
χ^2	4217.918	4649.052	4981.828

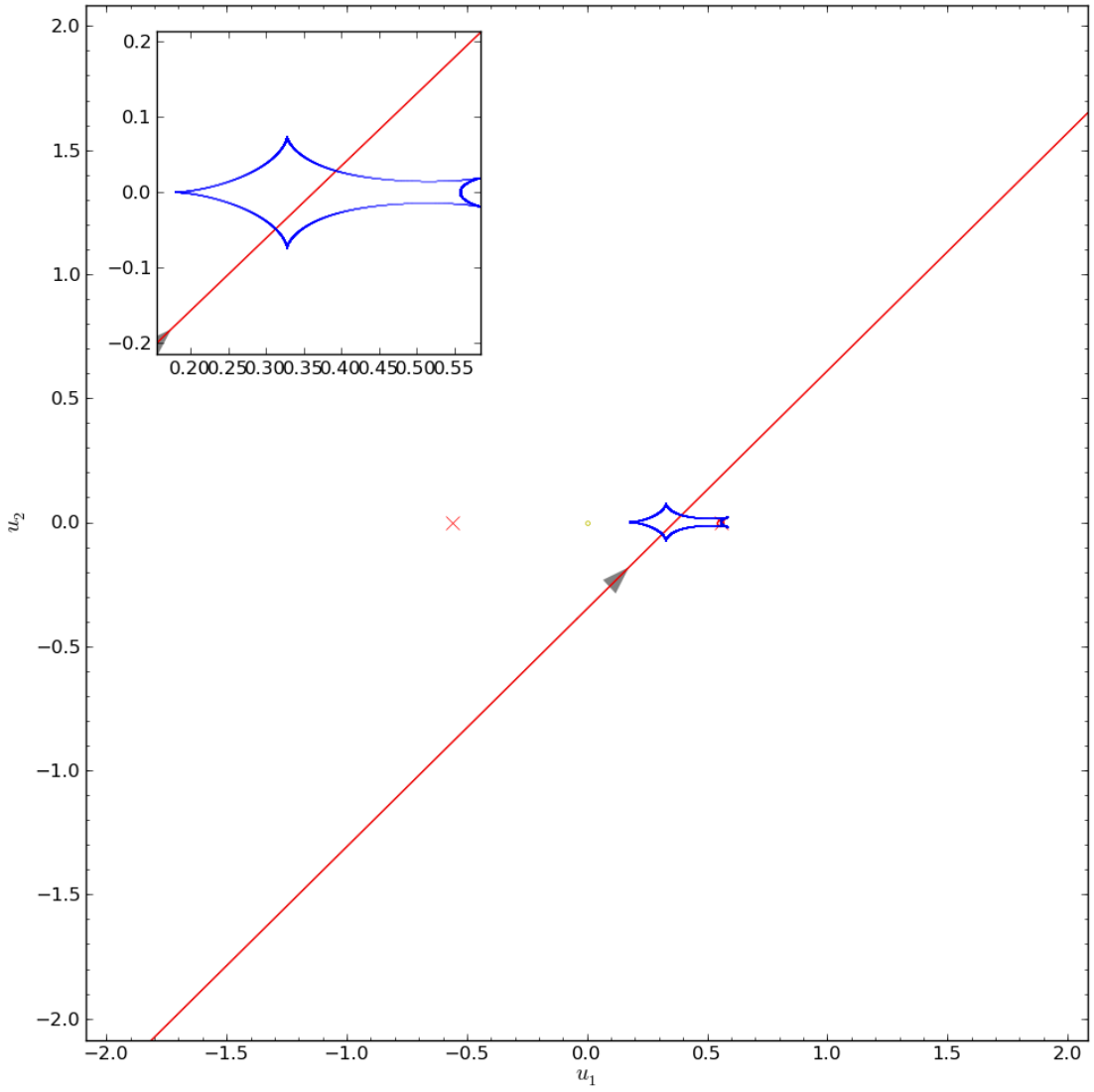


FIGURE 7.61: A caustic map with overlaid trajectory for the Bond et al. (2004) planetary model (Table 7.18).

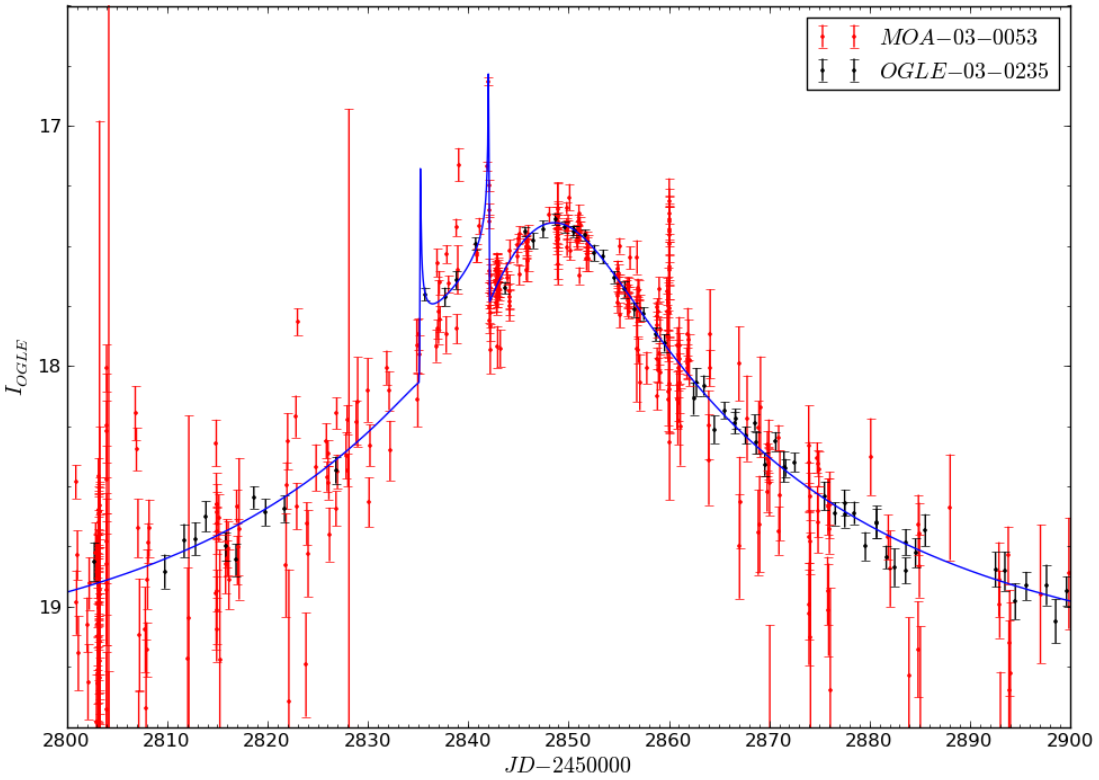


FIGURE 7.62: A Light curve of Bond et al. (2004) planetary model (Table 7.18).

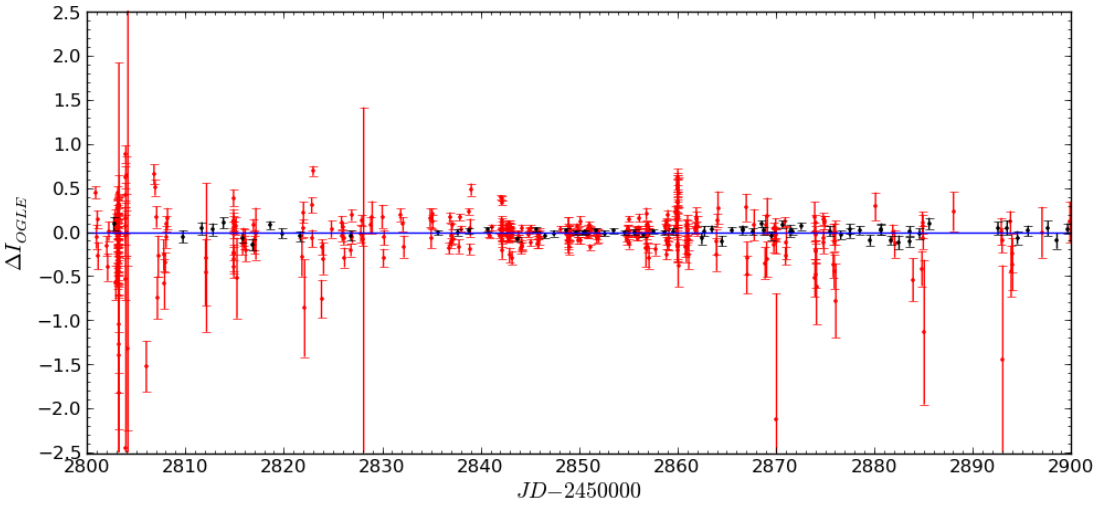


FIGURE 7.63: Light curve residuals of Bond et al. (2004) planetary model (Table 7.18).

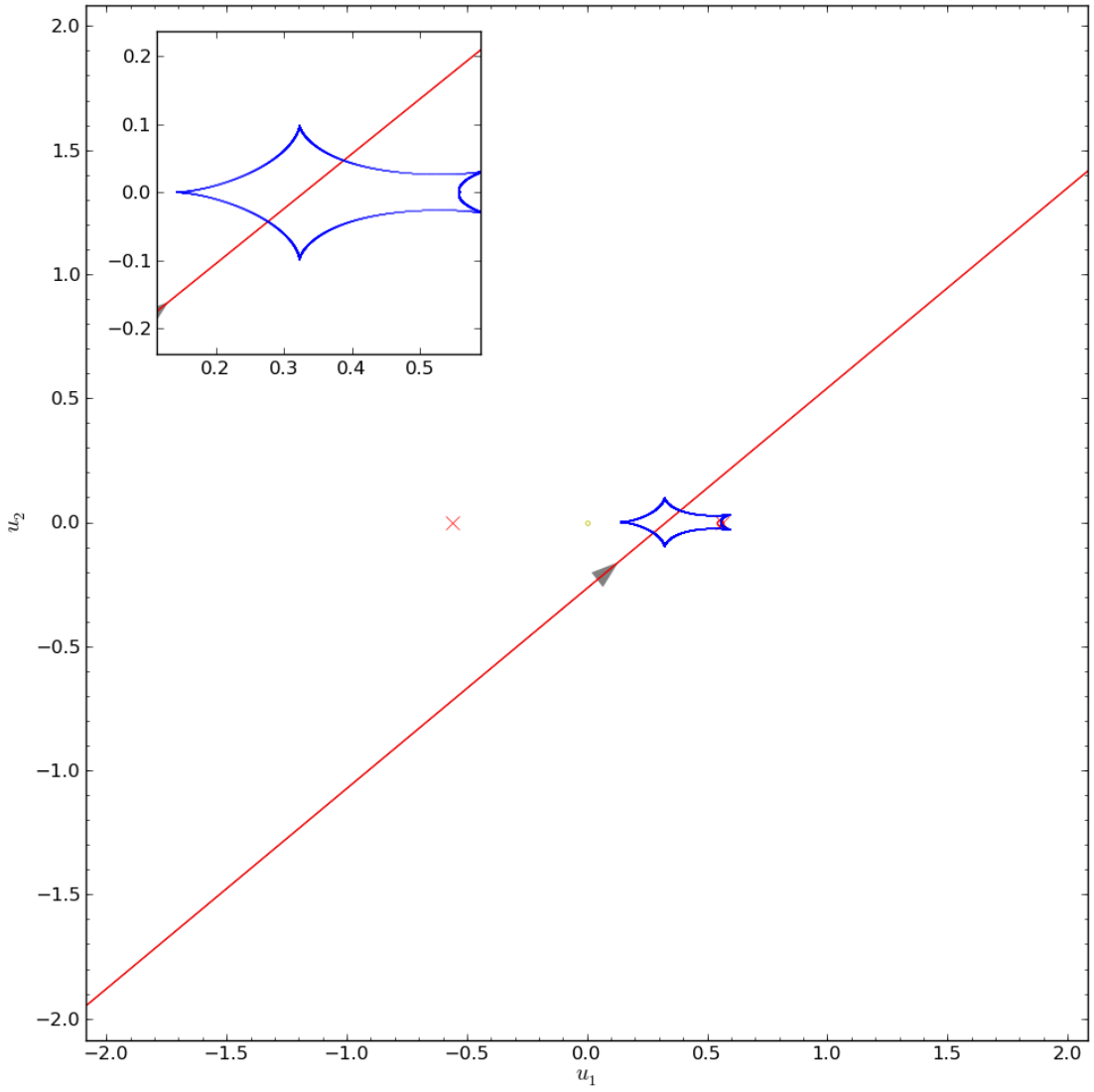


FIGURE 7.64: A caustic map with overlaid trajectory for the Bond et al. (2004) alternate planetary model (Table 7.18).

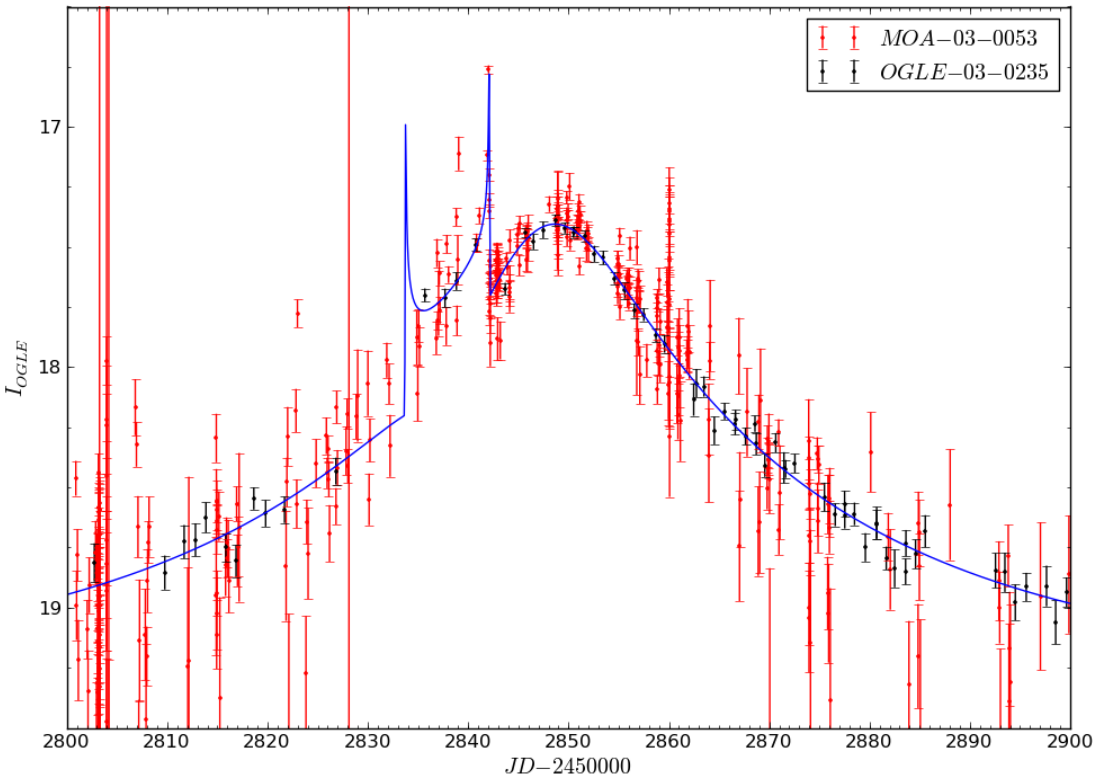


FIGURE 7.65: A Light curve of Bond et al. (2004) alternate planetary model (Table 7.18).

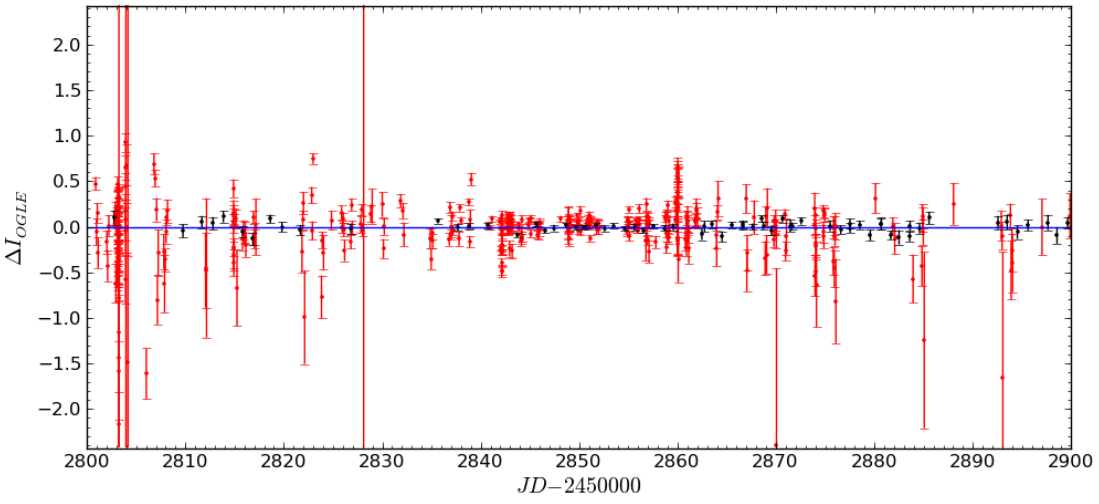


FIGURE 7.66: Light curve residuals of Bond et al. (2004) alternate planetary model (Table 7.18).

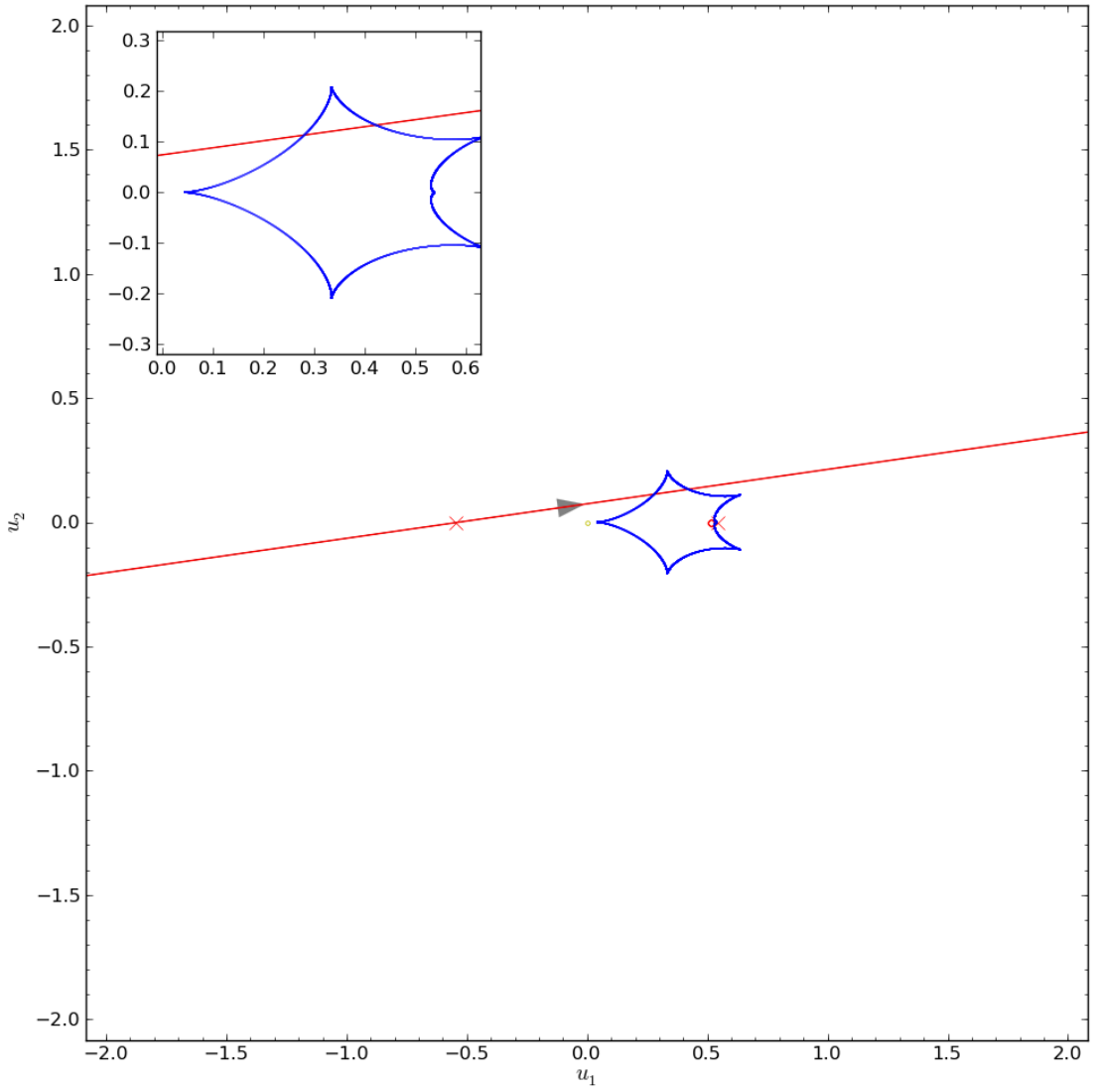


FIGURE 7.67: A caustic map with overlaid trajectory for the Bond et al. (2004) non-planetary model (Table 7.18).

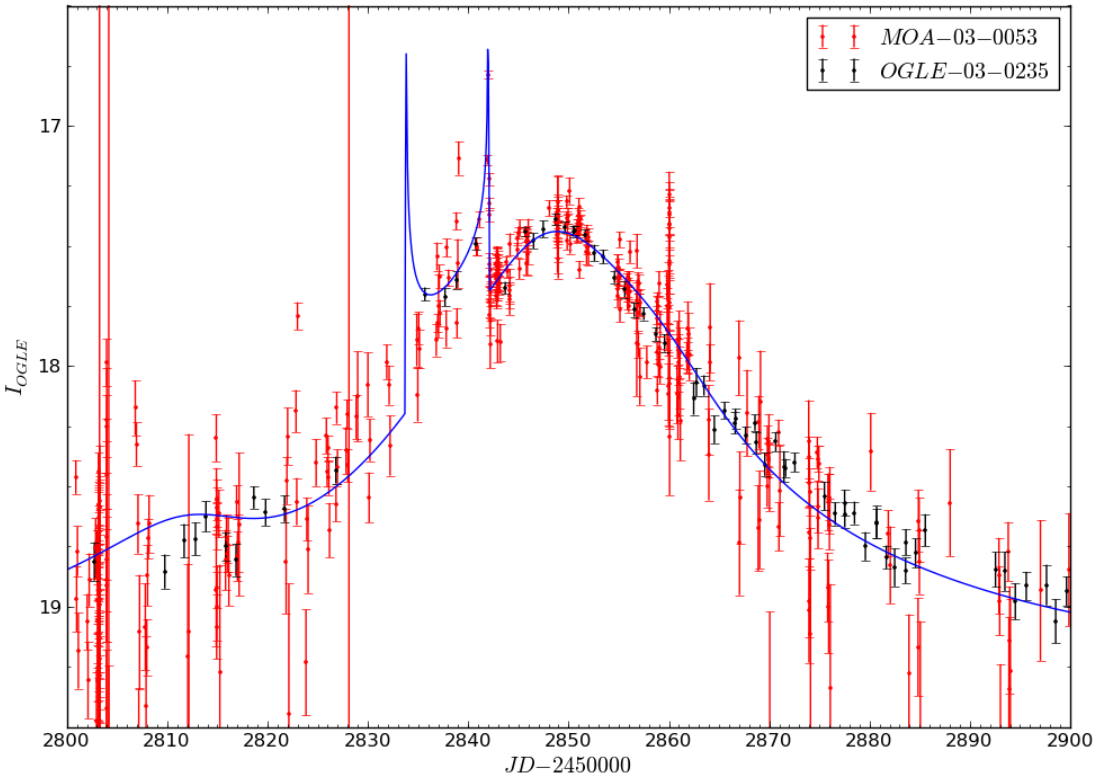


FIGURE 7.68: A Light curve of Bond et al. (2004) non-planetary model (Table 7.18).

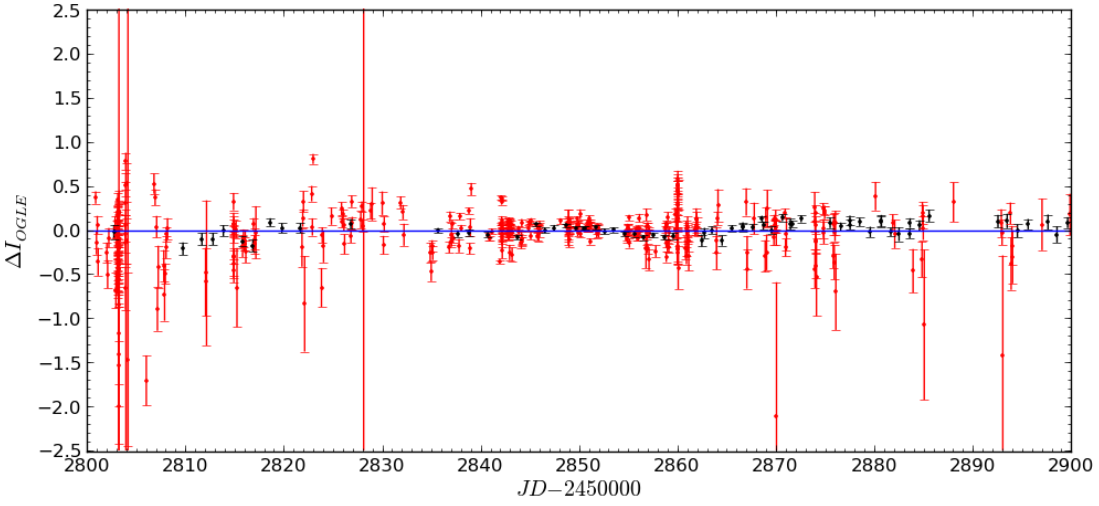


FIGURE 7.69: Light curve residuals of Bond et al. (2004) non-planetary model (Table 7.18).

To model this event the data was taken from each survey telescope's online data archive with no additional modifications made, which replicates how the modelling system would deal with this event if it was ongoing. Data in the AJD range of 2300-3400 is used to model the event, which contains 765 MOA and 274 OGLE data points. An input of a minimum possible magnitude difference of 2.0 is provided to the model requirements, as well as the model needing to contain three peaks, where two are located around 2848.675 (the caustic exit) and 2848.7 (the central peak). Using these input parameters a grid search is performed in 48 minutes, producing the following minimized χ^2 map (Figure 7.70).

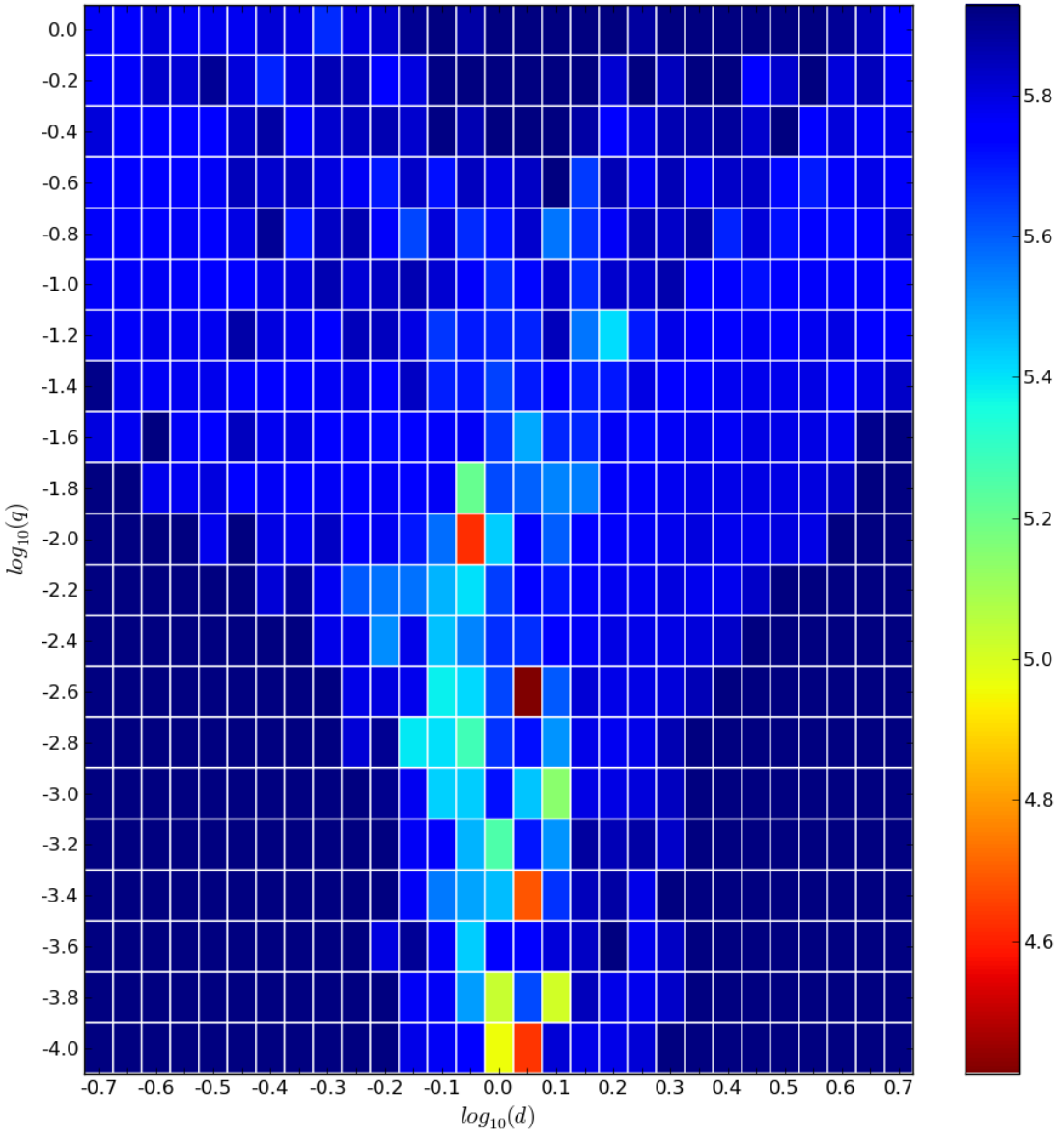


FIGURE 7.70: Minimised χ^2 d, q initial grid search map of OGLE-2003-BLG-0235

This lead to the further analysis of the areas of local minima, where six local MCMC

processes were initialized, producing the d, q movements shown overlaid on the χ^2 map (Figure 7.71), and the change in the trajectories over the caustic structures shown in Figure 7.72, resulting in the parameter sets shown in Table 7.19.

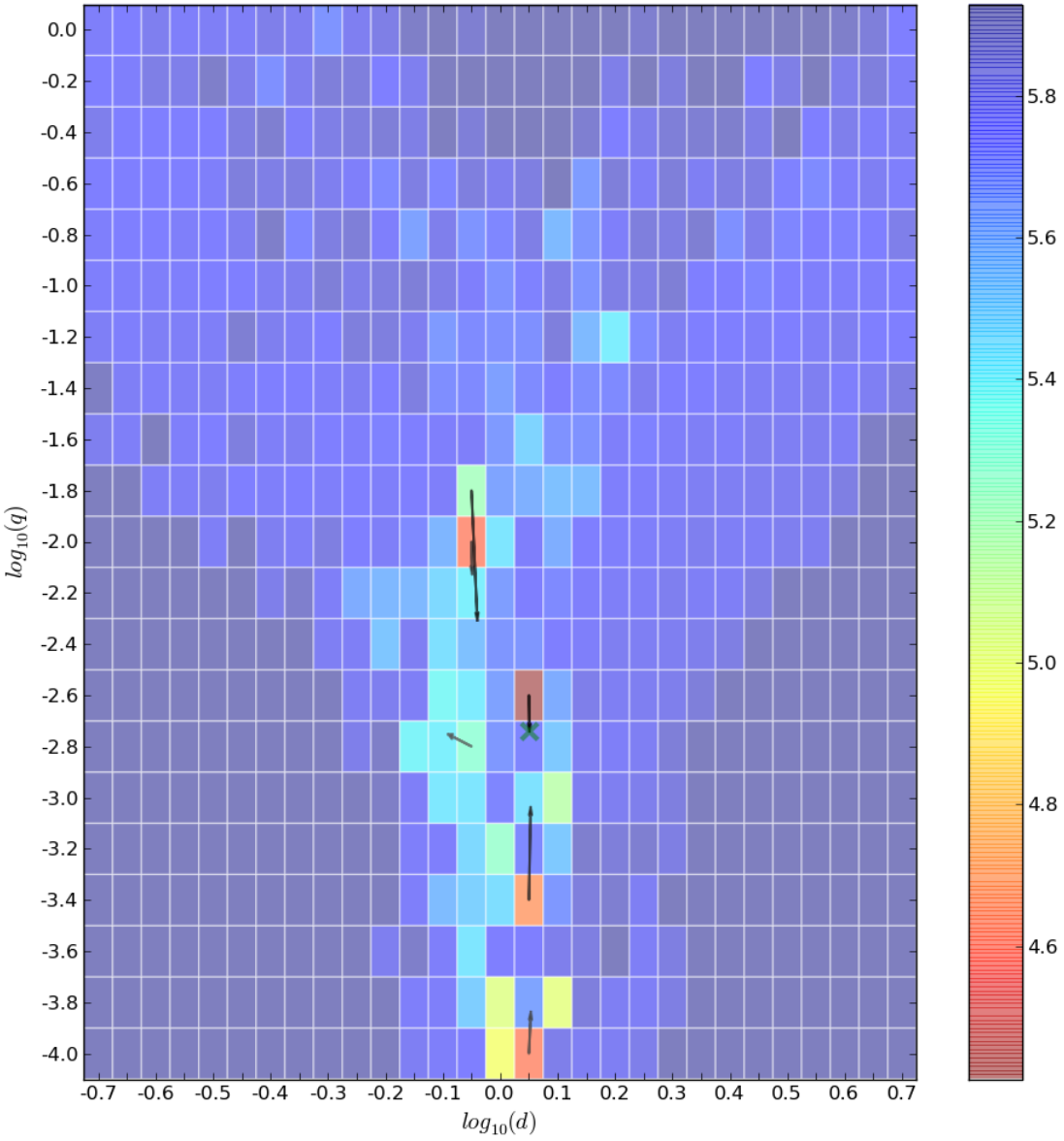


FIGURE 7.71: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2003-BLG-0235.

Of these six local MCMC searches, the one which resulted in the lowest χ^2 solution was determined to be in the global minimum. The final parameter set from this chain is used to normalise the data's errors before a final EMCEE process is performed. This explores the local area very thoroughly, and determines the parameter and error term for each parameter, as shown in Table 7.20.

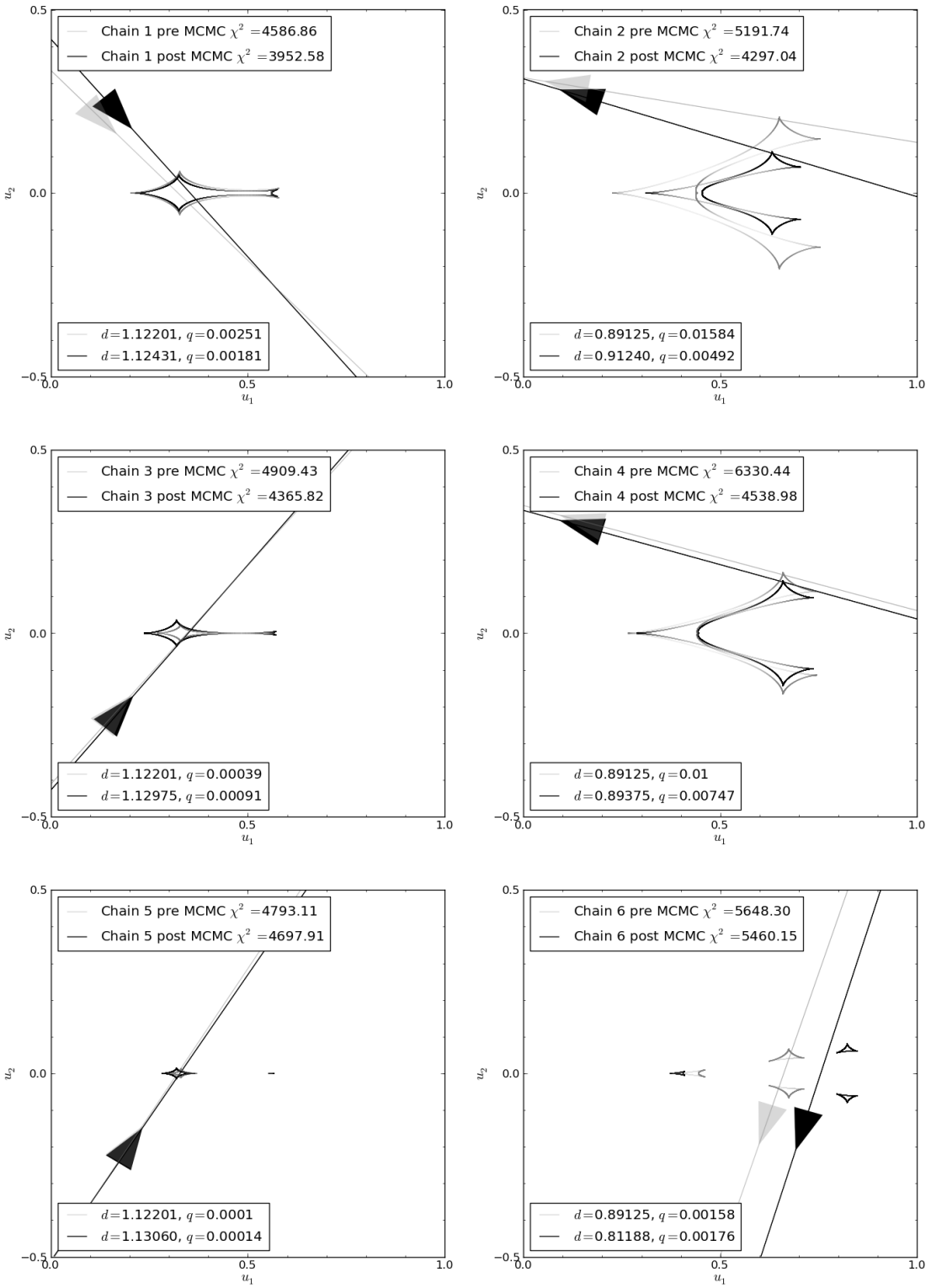


FIGURE 7.72: Local minima MCMC search result's change in caustic structure and trajectory.

TABLE 7.19: Seven parameter binary lens local minima model solutions of OGLE-2003-BLG-0235 MCMC search, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	1.12431	0.91240	1.12976	0.89375	1.13061	0.81189
q	0.00181	0.00493	0.00092	0.00748	0.00015	0.00177
ρ	9.81×10^{-4}	0.00103	0.00173	9.28×10^{-4}	1.09×10^{-4}	0.06135
u_0	-0.15789	-0.15828	0.16688	-0.19615	0.19977	0.33524
ϕ	5.41245	2.83053	0.88715	2.85364	1.00082	4.41716
t_0	2848.118	2849.590	2849.010	2849.858	2848.004	2847.366
t_E	55.224	54.73362	56.579	46.114	48.804	33.250
χ^2	3952.582	4297.042	4365.824	4538.988	4697.919	5460.158

TABLE 7.20: Seven parameter binary lens model EMCEE solution of OGLE-2003-BLG-0235, determined using error-normalized data.

Parameter	GPU solution
d	1.124 ± 0.038
q	0.00182 ± 0.00018
ρ	$0.0010 \pm 0.36 \times 10^{-4}$
u_0	-0.1583 ± 0.0040
ϕ	5.41 ± 0.18
t_0	2848.11 ± 0.27
t_E	55.53 ± 1.0

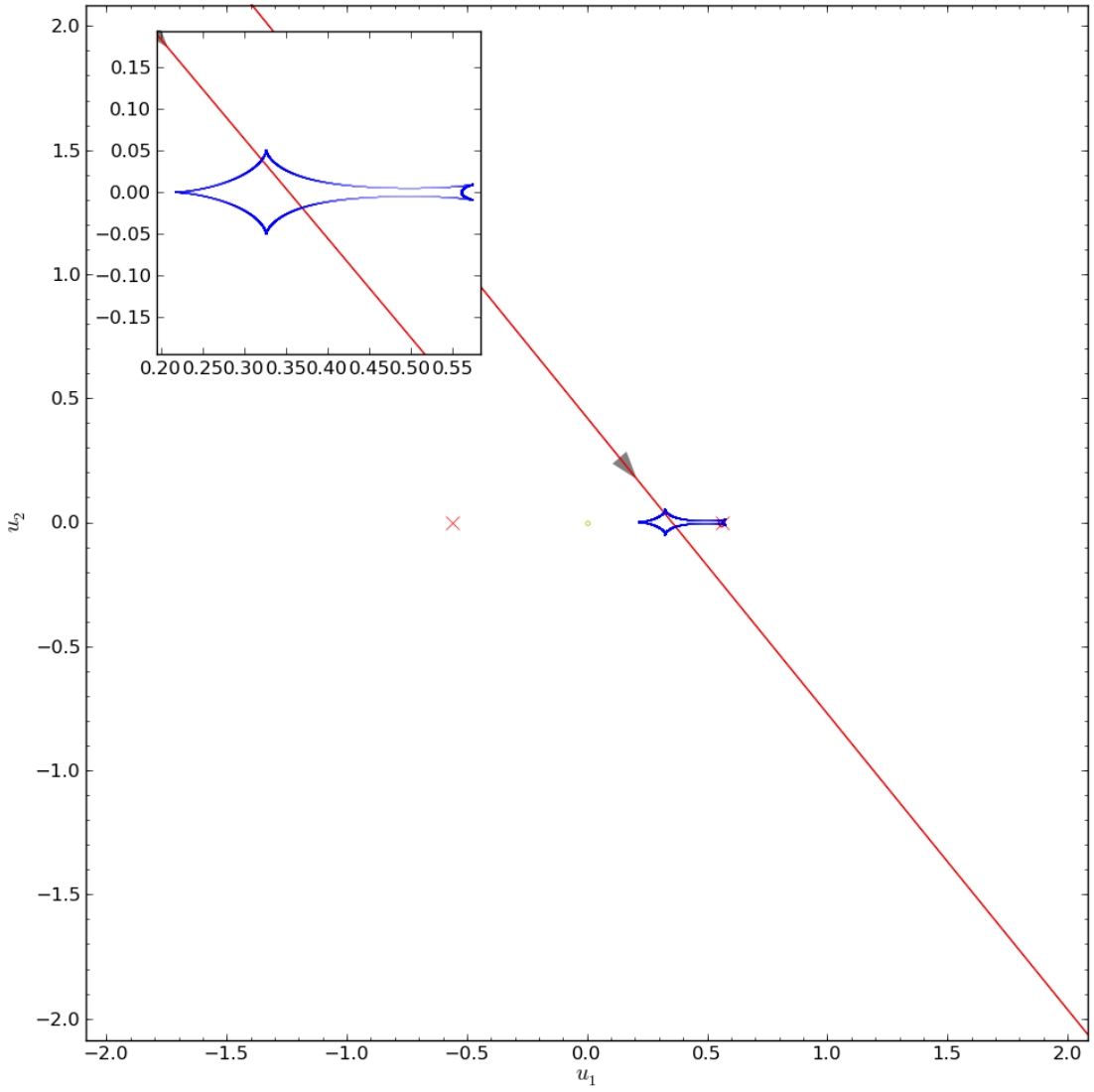


FIGURE 7.73: A caustic map with overlaid trajectory for the model solution (Table 7.20) determined by the GPU EMCEE method

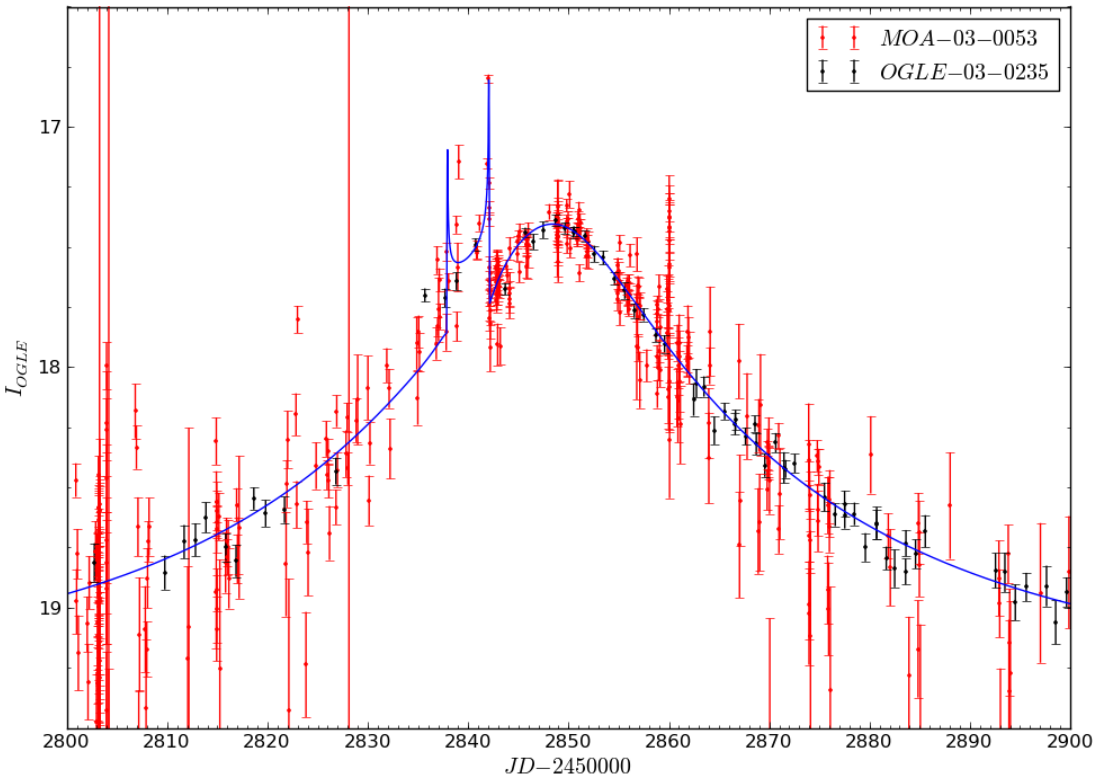


FIGURE 7.74: A Light curve of the model solution (Table 7.20) determined by the GPU EMCEE method

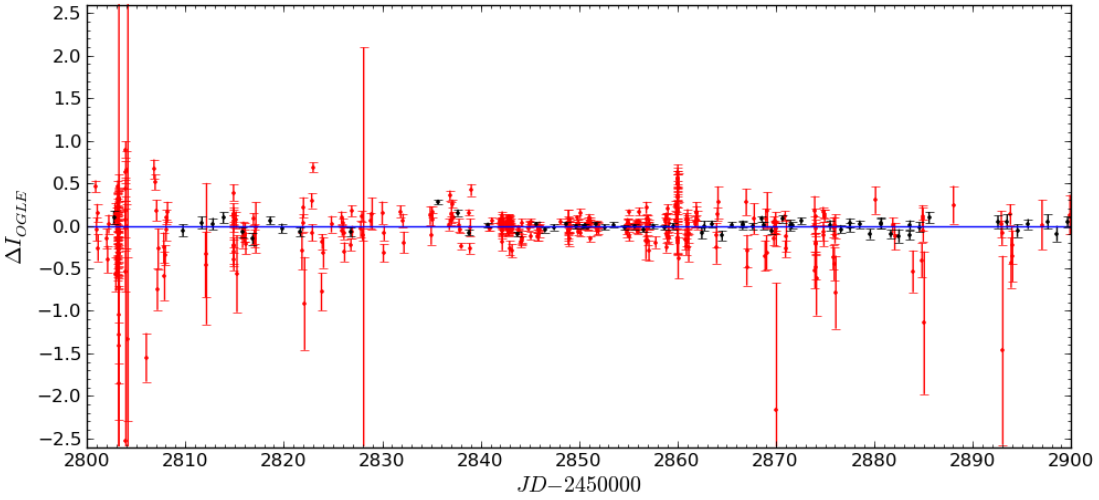


FIGURE 7.75: Light curve residuals of the model solution (Table 7.20) determined by the GPU EMCEE method

The results presented here successfully show the new GPU method rapidly locates the same global minima as previously published results. Minor discrepancies exist between the solutions' parameter values, however, the calculated χ^2 values shows the GPU solution presented here to be favorable. This may results from alternate data being used, possibly from re-reduced data or unique treatments that are not detailed in the published papers. Bennett (2010) not only showed agreement between the model solutions, but he also quoted a time taken to model the event. Using his method took 5 hours and 14 minutes, which involved searching approximately 70,000 solutions in a restricted parameter space around the areas where the solutions are known to lie. In comparison, the GPU method presented here took 3 hours and 54 minutes to locate the global minimum, exploring at least 8,221,202 up to a maximum of 4,098,625,000 different parameter solutions, in an unrestricted parameter space. This does not include the additional time spent at the end with normalized error data to determine the error estimates on each parameter, which can sometimes take just as long as the initial search.

7.4 OGLE-2012-BLG-0406

The OGLE IV survey detected a microlensing event in 2012 April 17 which it designated OGL-2012-BLG-0406. Due to the expected low maximum magnification it was a low priority target until 2012 July 1 when the magnification started to rise rapidly, deviating from the point-source model. This deviation was determined to be caused by a binary lens, resulting in many follow up observations from additional telescopes. Four days later an announcement was made (C. Han, e-mail to ulens-analysis mail list on 2012 July 5 AJD 6113.95) that the previous deviation was likely caused by a planetary system and that an additional caustic feature was expected very soon. This prompted in intense follow up observations from a large group of telescopes. After the predicted planetary feature passed, high cadence observations continued for the duration of the microlensing event. This event has been thoroughly studied and modeled by a few groups, most notably Poleski et al. (2014), who modeled the event using only OGLE survey telescope data and Tsapras et al. (2014) who used data from ten separate observing sites. Y. Tsapras has kindly provided the data and further details of the model used in his publication (but not his error rescaling values), allowing for a partial model comparison to be made.

Y. Tsapras began his parameter space search by considering the single lens model that defines the general shape of the data, if no anomaly existed. Due to the deviations in the data being of short time scale on the shoulder of a Paczyński curve, it is safe to conclude that the cause of the anomaly is from the source passing close to a central caustic. In

such a case, the single lens parameters of u_0, t_0, and t_E (which can be determined with relative ease) are similar to the matching parameters of the binary lens model. Tsapras et al. (2014) uses this information to perform a hybrid grid search of d, q, and ϕ, where u_0, t_0, t_E, and ρ are minimized using MCMC methods. The grid is searched between the limits of $-1 \leq \log_{10}(d) \leq 1$, $-5 \leq \log_{10}(q) \leq 1$, and $0 \leq \phi \leq 2pi$ with the aim to locate all areas of local minimum, before they are further refined by narrowing the grid search parameter space. Finally a χ^2 optimization in all seven parameters is performed in each local minimum to determine the global minimum. This search resulted in Y. Tsapras identifying the binary lens model solution shown in Table 7.21.

R. Poleski used a similar initial search routine, with a few differences in the filtering and error scaling of the data used, and an assumed limb-darkening coefficient of $\Gamma_I = 0.353$ (as opposed to Y. Tsapras who used $\Gamma_I = 0.53$). R. Poleski found a very similar parameter set, shown in Table 7.22. Both Y. Tsapras and R. Poleski came to the same conclusions with only minor differences in their results caused by the different data sets used in their modelling.

TABLE 7.21: A seven parameter set for the standard binary lens model solution presented by Y. Tsapras of event OGLE-2012-BLG-0406, using data from 10 different observing sites (between AJD 5261.84 and 6499.56) and an assumed linear limb darkening coefficient of $\Gamma_I = 0.53$, with no higher order affects included.

Parameter	Model
d	1.346 ± 0.001
q	0.00533 ± 0.00003
ρ	0.01105 ± 0.00005
u_0	0.530 ± 0.001
ϕ	0.526 ± 0.001
t_0	6141.47 ± 0.02
t_E	62.34 ± 0.05
χ^2	6525.346

TABLE 7.22: A seven parameter set for the standard binary lens model solution provided by R. Poleski of the event OGLE-2012-BLG-0406, using only OGLE data (between AJD 5261.84 and 6499.56), and an assumed linear limb darkening coefficient of $\Gamma_I = 0.353$, with no higher order affects included.

Parameter	Model
d	1.3500 ± 0.0016
q	0.00578 ± 0.00008
ρ	0.0098 ± 0.0005
u_0	0.5425 ± 0.0022
ϕ	0.8653 ± 0.0016
t_0	6141.593 ± 0.031
t_E	62.63 ± 0.16
χ^2	2678.785

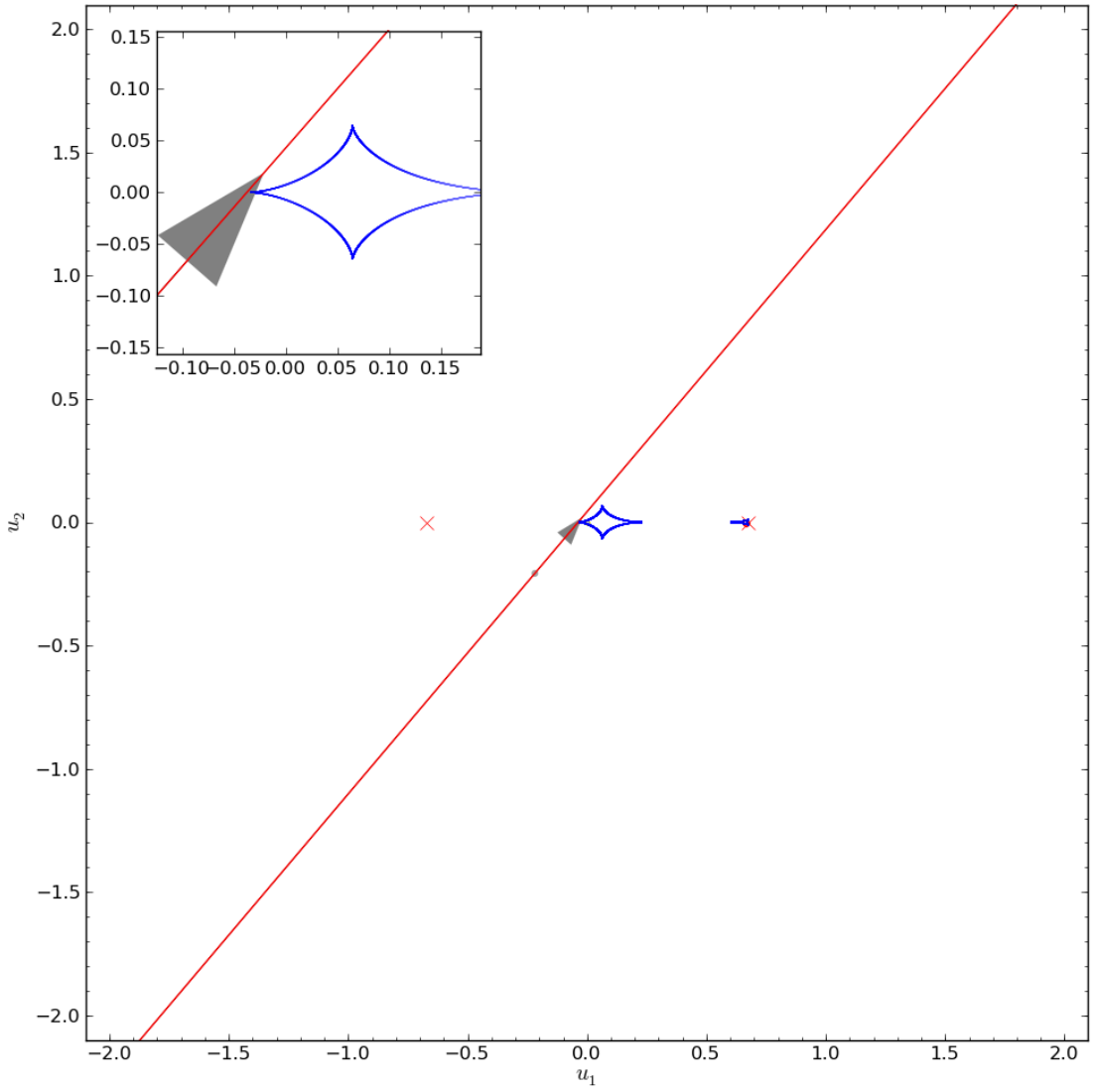


FIGURE 7.76: A caustic map with overlaid trajectory for Y. Tsapras' planetary model (Table 7.21).

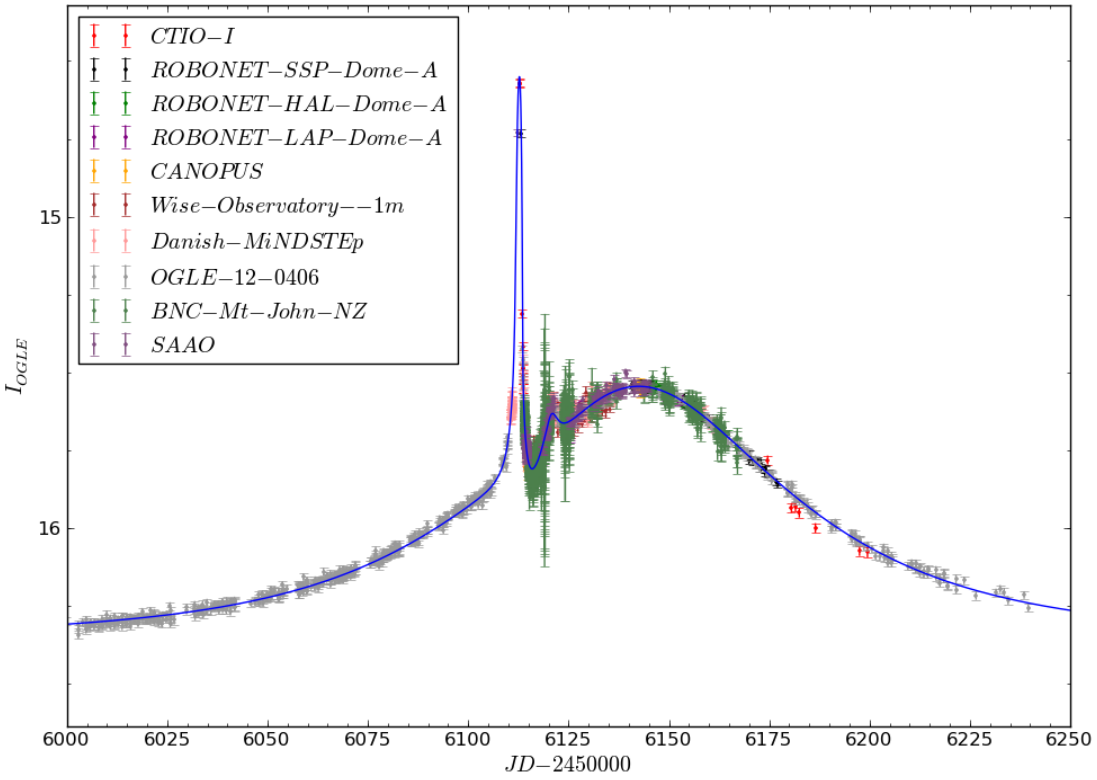


FIGURE 7.77: A Light curve of Y. Tsapras' planetary model (Table 7.21).

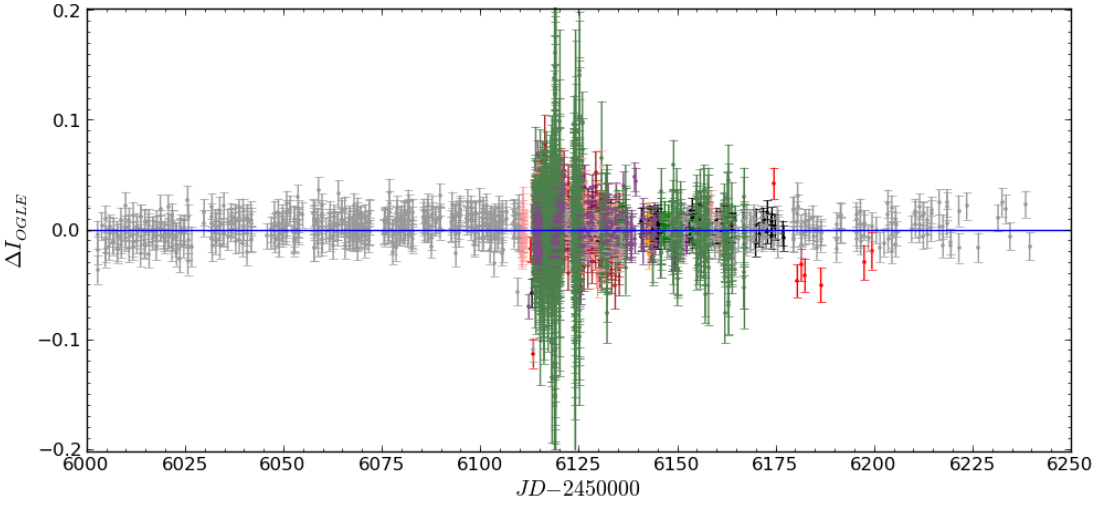


FIGURE 7.78: Light curve residuals of Y. Tsapras' planetary model (Table 7.21).

Using the GPU modelling approach outlined in section 5.6 a seven parameter binary lens search was performed, first using only the OGLE parameter set (3013 data points). The initial search successfully located well-defined areas of local minimum in the planetary region expected, shown in Figure 7.79, which was further explored using local MCMC methods. As described earlier this search process ensures possible close and wide degenerate solutions are explored by using several MCMC searches starting at the best close and wide solutions. All chains show movement towards their common local minima, seen in Figure 7.80, with the wide solution providing a significantly better χ^2, as detailed in Table 7.23.

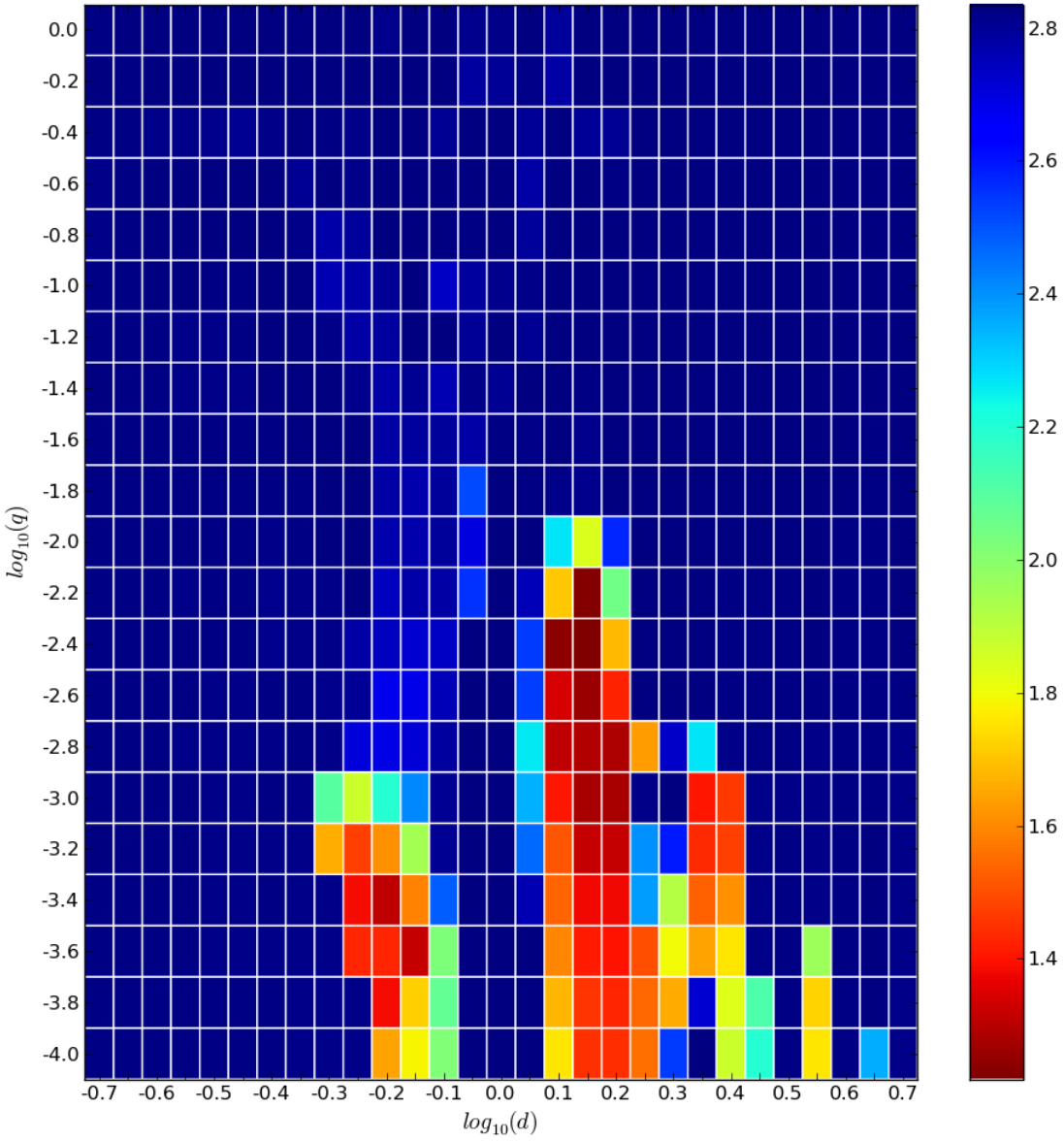


FIGURE 7.79: Minimised χ^2 d, q initial grid search map of OGLE-2012-BLG-0406

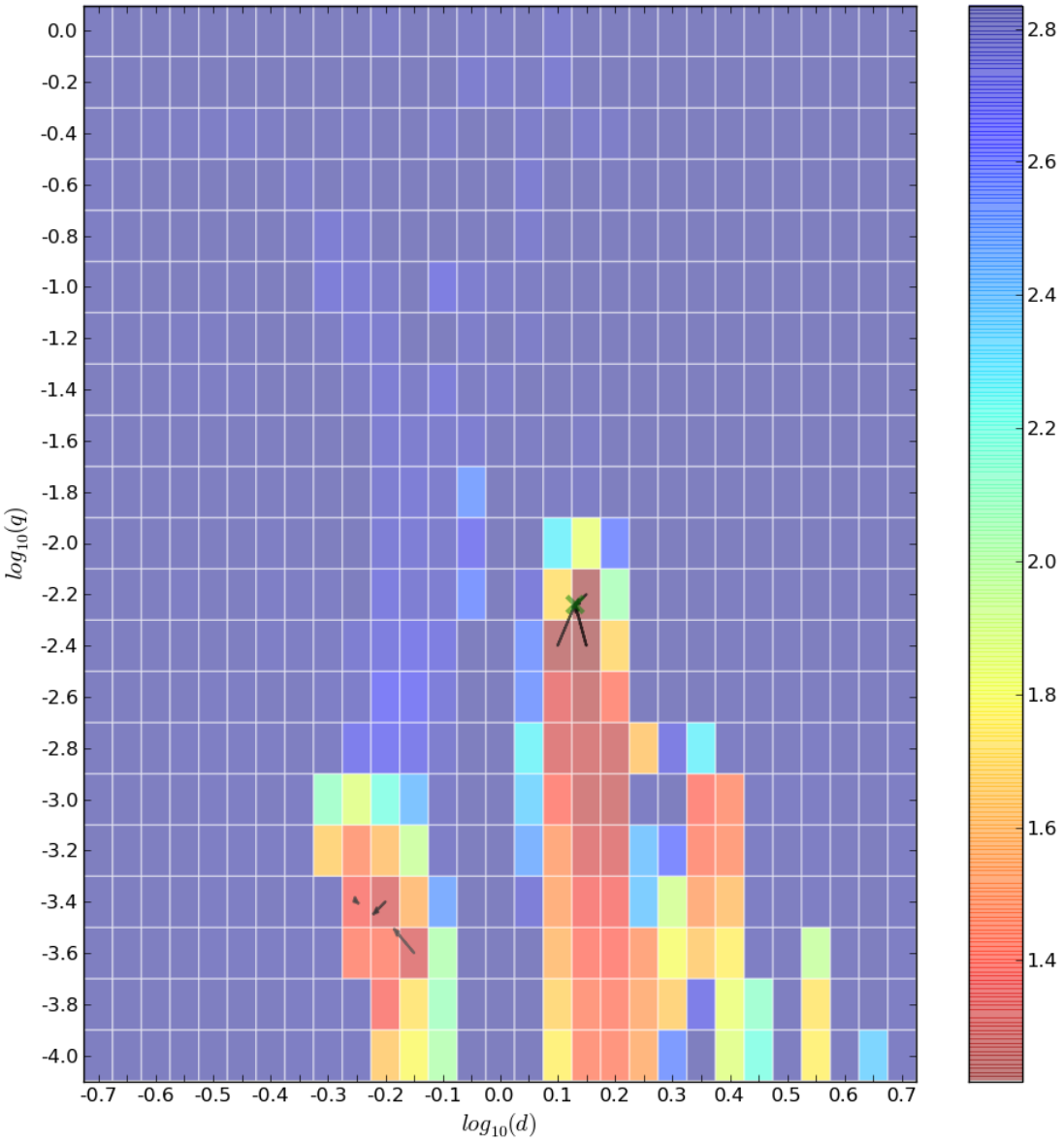


FIGURE 7.80: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q χ^2 grid search map of OGLE-2012-BLG-0406.

TABLE 7.23: A seven parameter set for the standard model solution produced by the GPU search method of the event OGLE-2012-BLG-0406, using only OGLE data (between AJD 5261.84 and 6499.56), and a limb darkening coefficient of $\Gamma_I = 0.53$.

Parameter	Model
d	1.35 ± 0.13
q	0.00576 ± 0.0151
ρ	0.0098 ± 0.0013
u_0	-0.539 ± 0.044
ϕ	5.419 ± 0.925
t_0	6141.437 ± 0.451
t_E	62.52 ± 2.34
χ^2	2481.512

Using only the OGLE data and an assumed linear limb darkening coefficient of $\Gamma_I = 0.53$, the GPU search method successfully located the same global minimum as R. Poleski and Y. Tsapras to a 1σ accuracy. This implies that a full parameter search including all data sources is likely to locate the same global minimum due to the well sampled OGLE data. By using all data sources (81 from CTIO I band Chile, 181 from Robonet Siding Spring Dome A, 83 from Robonet Haleakala Dome A, 131 from Robonet La Palma Dome A, 210 from Canopus Observatory Tasmania, 180 from Wise Observatory Israel, 473 from Danish MiNDSTEp Chile, 3013 from OGLE Chile, 1856 from Boller and Chivens New Zealand, and 226 from SAAO Elizabeth Telescope South Africa), the computation time takes longer, but provides a more constrained solution due to the large number of additional observations. The full search with all data sets produced a very similar minimized χ^2 d, q map as to the one produced with only OGLE data. Furthermore, the local MCMC part of the search process also reproduced a similar map, Figure 7.81, with the six parameter sets shown in Table 7.24. Choosing the lowest χ^2 parameter set from this list, an EMCEE search using normalised data lead to the parameter solution shown in Table 7.25, which produces the caustic structure, light curve and residuals shown in Figures 7.82-7.84.

The analysis of this event (OGLE-2012-BLG-0406) using the GPU search methods presented in this thesis is shown to successfully identify the same global solution as has been previously found in the publications of Poleski et al. (2014) and Tsapras et al. (2014).

TABLE 7.24: Seven parameter Binary lens local minima model solutions of OGLE-2012-BLG-0406, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	1.3465	1.36297	1.79690	0.51469	0.59626	0.48294
q	0.00538	0.00609	0.00094	0.000667	0.000650	0.00070
ρ	0.01079	2.53×10^{-8}	0.00545	0.01919	0.00980	0.01100
u_0	−0.53001	−0.55926	−1.04353	1.22359	0.89214	1.37530
ϕ	5.43119	5.41106	5.30295	4.23747	4.18512	4.25467
t_0	6141.495	6141.420	6141.279	6141.186	6141.312	6141.064
t_E	62.2961	61.29593	42.017	38.638	46.892	35.589
χ^2	6356.392	7403.782	14386.986	21698.705	21241.881	20258.229

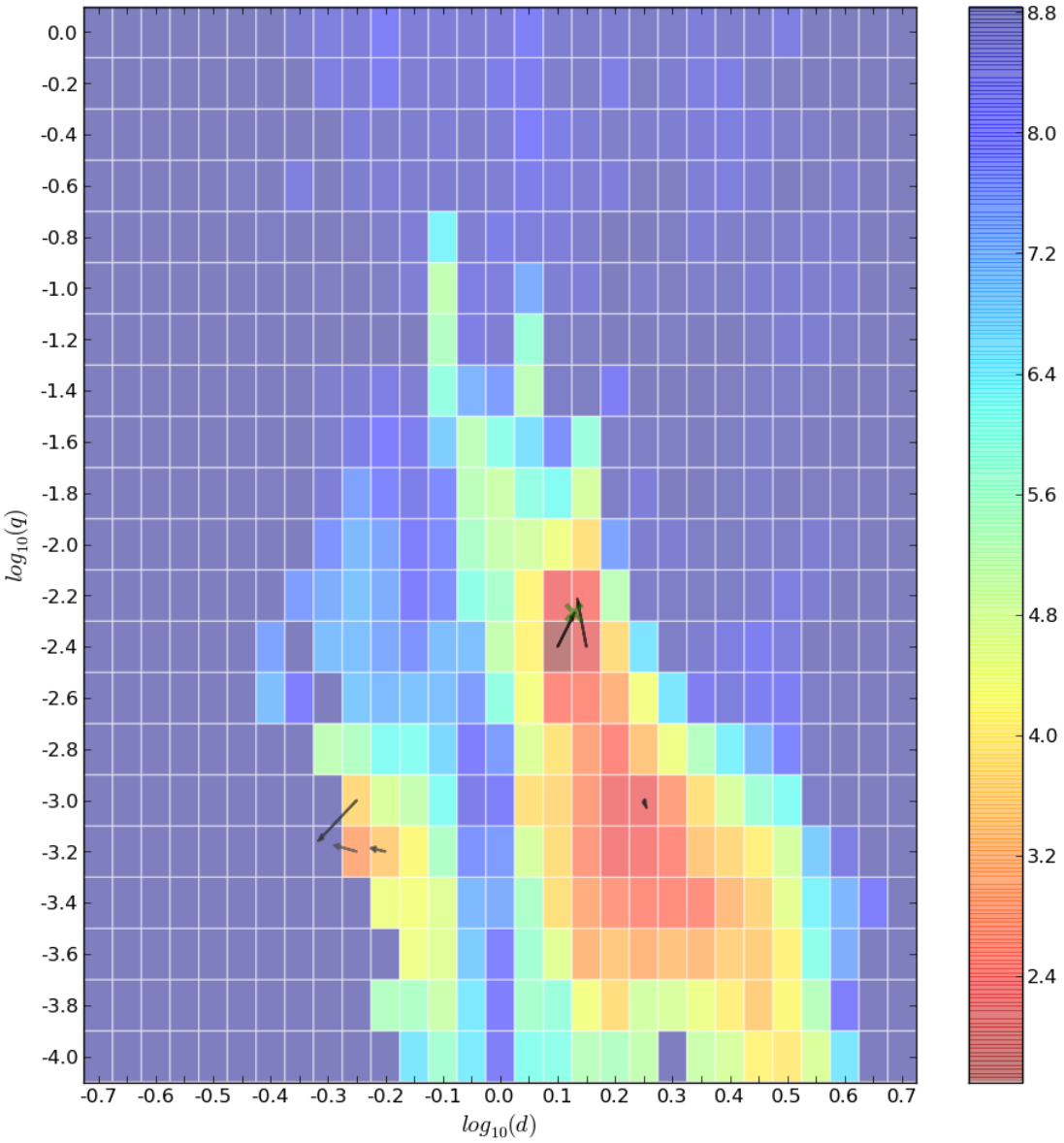


FIGURE 7.81: All data local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q grid search map of OGLE-2012-BLG-0406.

TABLE 7.25: A seven parameter binary lens model solution determined with the GPU search method of the event OGLE-2012-BLG-0406, using all data (between AJD 5261.84 and 6499.56), and a limb-darkening coefficient of $\Gamma_I = 0.53$.

Parameter	Model
d	1.35 ± 0.11
q	0.00535 ± 0.00044
ρ	0.0108 ± 0.0007
u_0	-0.532 ± 0.037
ϕ	5.43 ± 0.97
t_0	6141.47 ± 0.65
t_E	62.5 ± 2.3

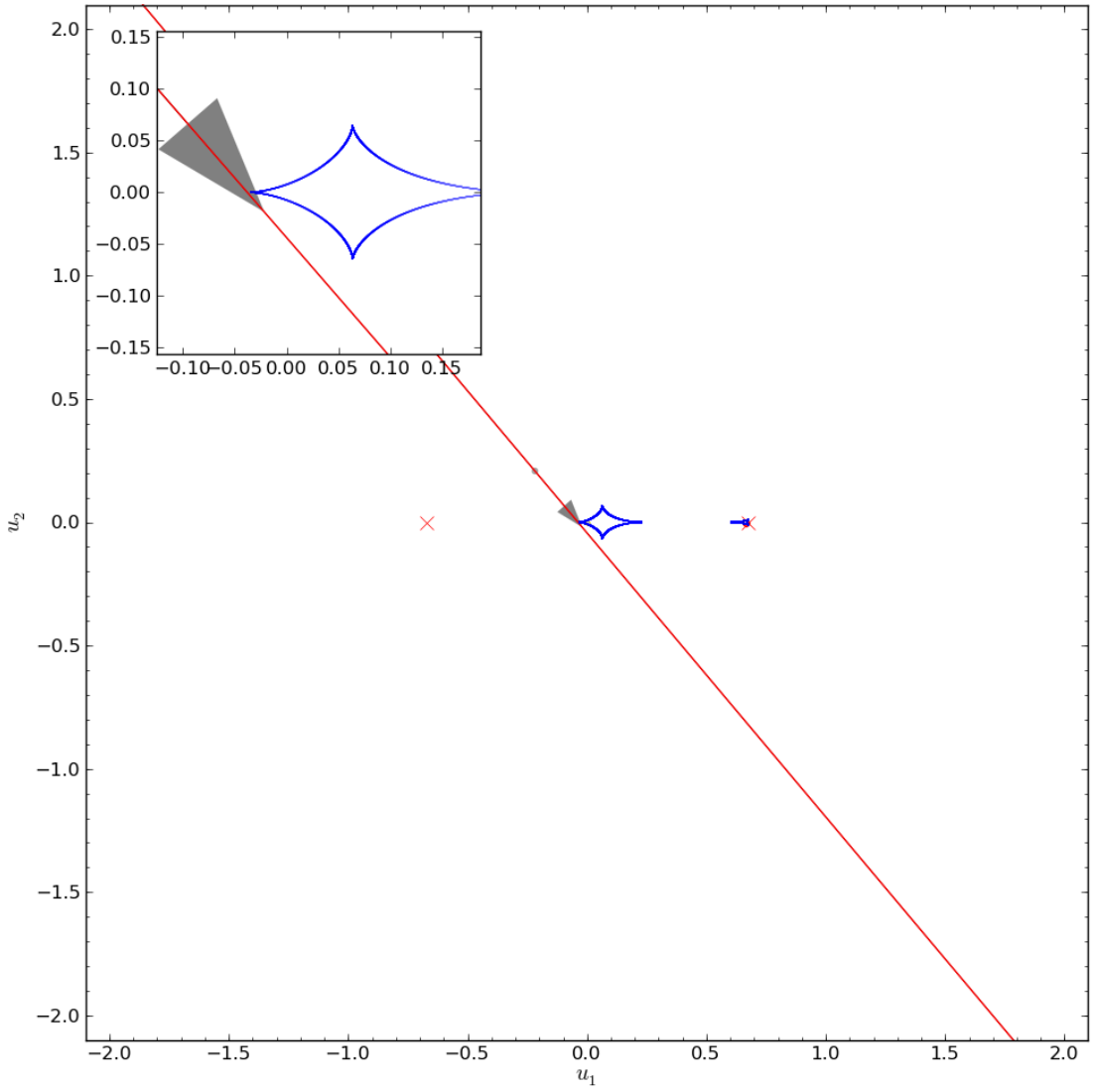


FIGURE 7.82: All data EMCEE solution's (Table 7.25) caustic trajectory of OGLE-2012-BLG-0406.

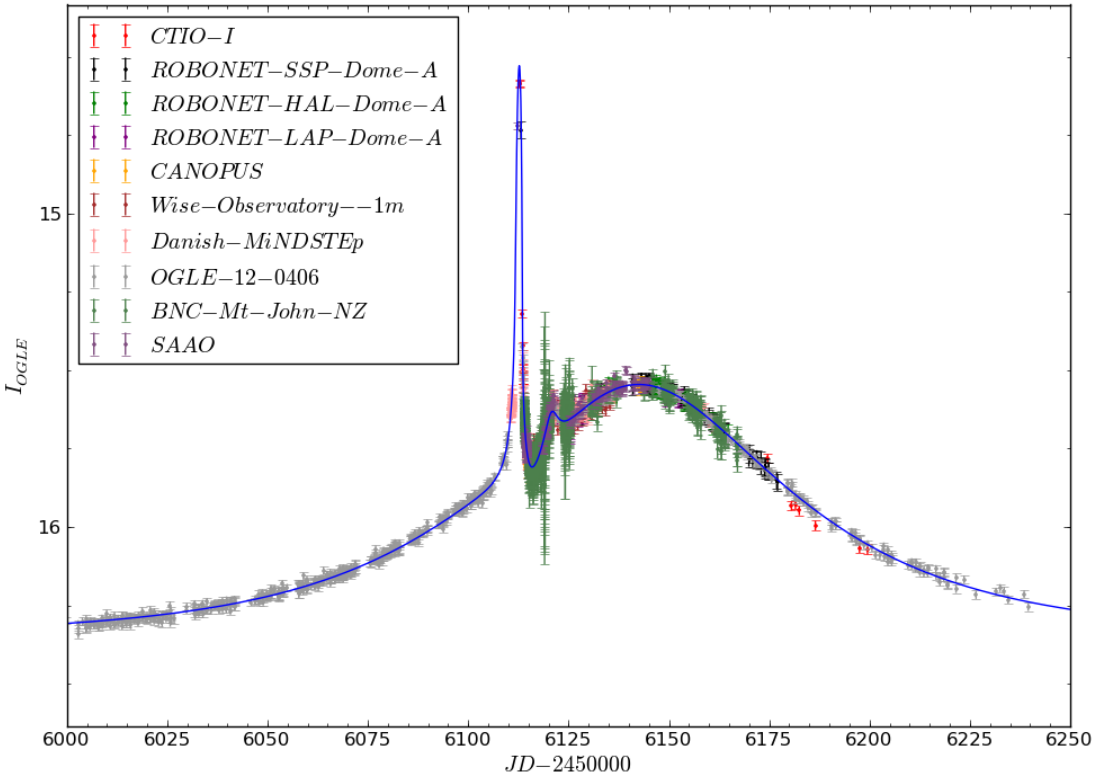


FIGURE 7.83: All data EMCEE solution's (Table 7.25) light curve of OGLE-2012-BLG-0406.

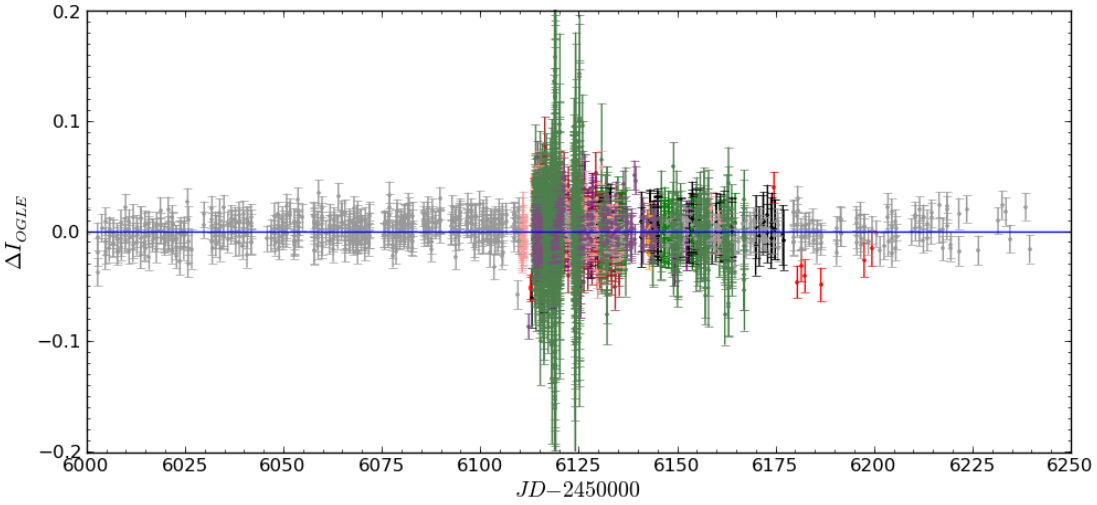


FIGURE 7.84: All data EMCEE solution's (Table 7.25) light curve residuals of OGLE-2012-BLG-0406.

7.5 Summary

The search method detailed in earlier chapters has been used in the modelling of sixteen past events, all of which have previously been modelled with published articles detailing the solutions found. From the modelling of all these events it is shown that fourteen located the same global minimum, occasionally with a close/wide degenerate solution, and in three cases (OGLE-2002-BLG-0114, OGLE-2003-BLG-0170, OGLE-2003-BLG-0291) the GPU search method located a better solution. In one case (OGLE-2003-BLG-0124) an equally likely but different solution was identified.

Only once (OGLE-2003-BLG-0056) did the GPU method locate a solution that was not as good as the presented solution in the published article. This situation was investigated, identifying that the global solution was located but not explored further due to the limitation in the number of local area MCMC searches. A larger number of MCMC search chains would have explored this solution and found the global minimum, however, it would have come at the cost of increased processing time. This situation highlights the risks taken when only considering the best three close and wide pixels of the grid search to explore further, which is used so the total search time can be kept down. Additional computational resources could help resolve this issue without a compromise to the total time required.

In one event (Section 7.3) a direct time comparison of our method with another modelling groups is presented. This shows that the GPU search method was able to locate the same global minimum 25% faster, despite searching at least 117 times (up to $\approx 58,000$ times) more parameter sets.

We have shown our method can correctly identify nearly all global minima present, and that it is very efficient. It confirms that our new method is a viable means of performing real-time analysis of binary lens events.

Chapter 8

Current binary lens results

8.1 Overview

Now that the model developed has been validated in Chapter 7, by confirming the global solutions of several previously published events, additional modelling of new events is continually being performed. Presented here is a selection of events from the 2014 observing season showing currently unpublished results for a set of binary and planetary systems. All dates and times are quoted in UTC.

Starting from the first detection of a microlensing event, the analysis system automatically retrieves the latest data from the survey telescope and any follow up observations made publicly available. A single lens model is performed and can be viewed through the web site. Analyzing a single lens model allows a visual inspection of the residuals to determine the presence of any possible deviations. If a potential deviation exists, the binary lens model search procedure is invoked. Using information from the single lens model such as its t_0 and t_E, or the number and location of peaks that exist in the data, the binary lens modeler can be started, and will perform a seven parameter χ^2 minimization search of a finite source binary lens model.

8.2 MOA-2014-BLG-0075/OGLE-2014-BLG-0241

On the 2014 March 21 we noticed a slight asymmetry in the single lens model of the event identified as MOA-2014-BLG-0075/OGLE-2014-BLG-0241. This prompted us to initialize our modelling procedure at AJD 6738.03. Included in the modelling of this event is a data set labeled by the OGLE EWS as OGLE-2014-BLG-0279. It is very similar to OGLE-2014-BLG-0241 with observations at identical AJDs, but with different

magnitudes. It was later identified by A.Udalski (A. Udalski, e-mail to ulens-analysis mail list on 2014 March 22 AJD 6738.93) that because the source is located between two close stars it lead to the EWS system alerting the same event twice. The modelling of the most recent 128 data points from each source was initialized by providing a lower limit for the maximum magnitude change of 5.0, and stating that the model was likely to be close to a Paczyński curve, therefore estimates of $t_E = 26.4$ and $t_0 = 6730.36$ were provided. After 3 hours and 17 minutes the grid search and local MCMC searches had identified the global minimum area shown in Figure 8.1. All chains converged to their appropriate local minima (3 close and 3 wide, shown in Table 8.1 and Figure 8.2), with all of the wide solutions providing lower χ^2 solution than any close solution. EMCEE was started from the lowest χ^2 chain of the wide solutions and determined the global solution shown in Table 8.2, which produces the caustic trajectory, light curve and residuals shown in Figures 8.3-8.5.

TABLE 8.1: Seven parameter binary lens local minima model solutions from a search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	1.28533	1.29824	1.26289	0.71451	0.72058	0.73106
q	0.00356	0.00390	0.00313	0.00812	0.00745	0.00704
ρ	0.00282	0.00299	0.00255	0.00583	0.00569	0.00535
u_0	−0.00460	−0.00477	−0.00431	−0.00690	−0.00674	−0.00630
ϕ	3.11365	3.11112	3.11920	2.95299	2.97132	2.96041
t_0	6730.748	6730.754	6730.732	6730.739	6730.741	6730.730
t_E	23.389	22.983	23.122	25.749	26.737	27.450
χ^2	978.153	978.430	979.793	1004.560	1006.287	1007.813

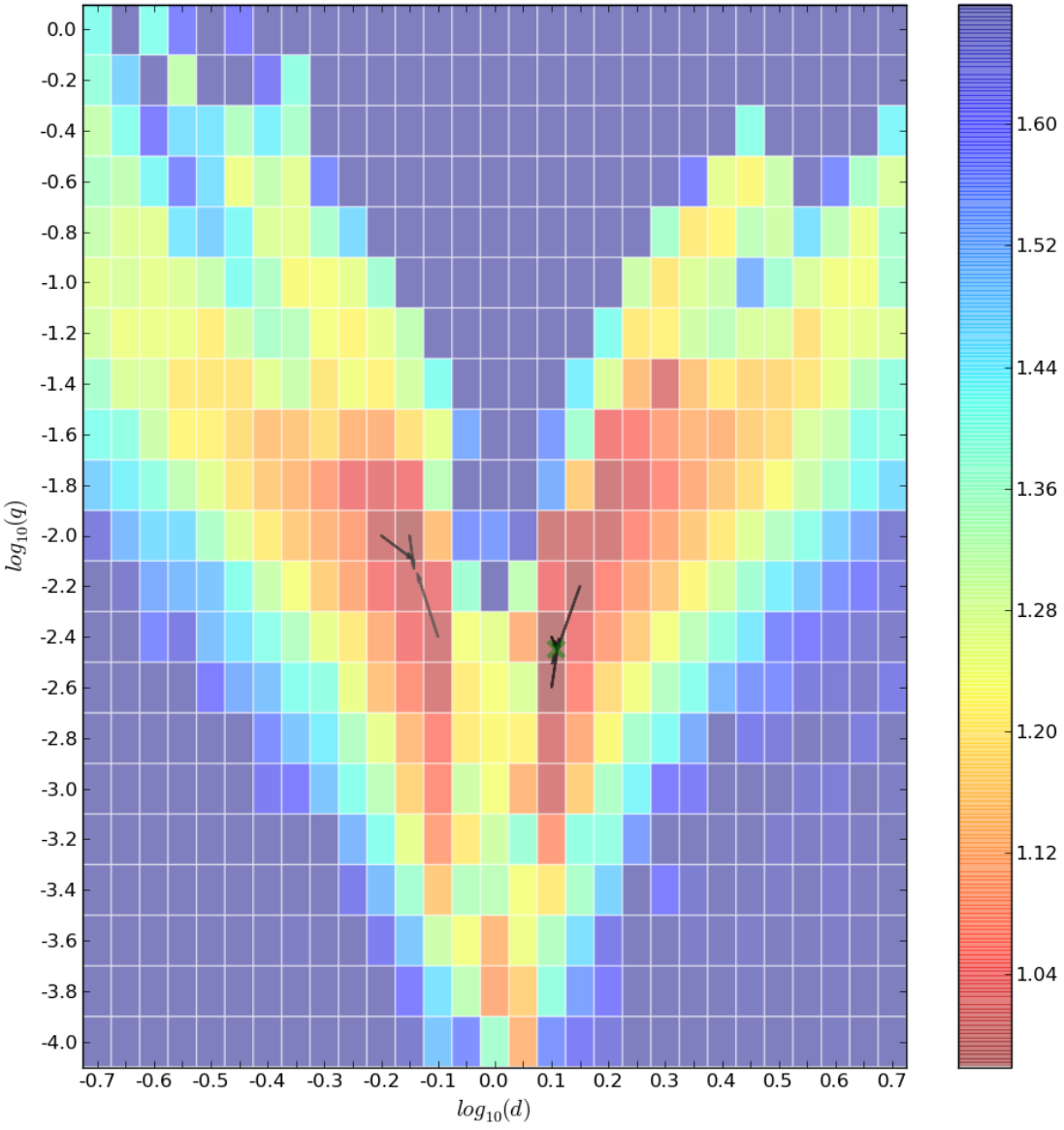


FIGURE 8.1: Local minima MCMC search (Table 8.1), showing d, q movement overlaid onto the initial minimized χ^2 d, q grid search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

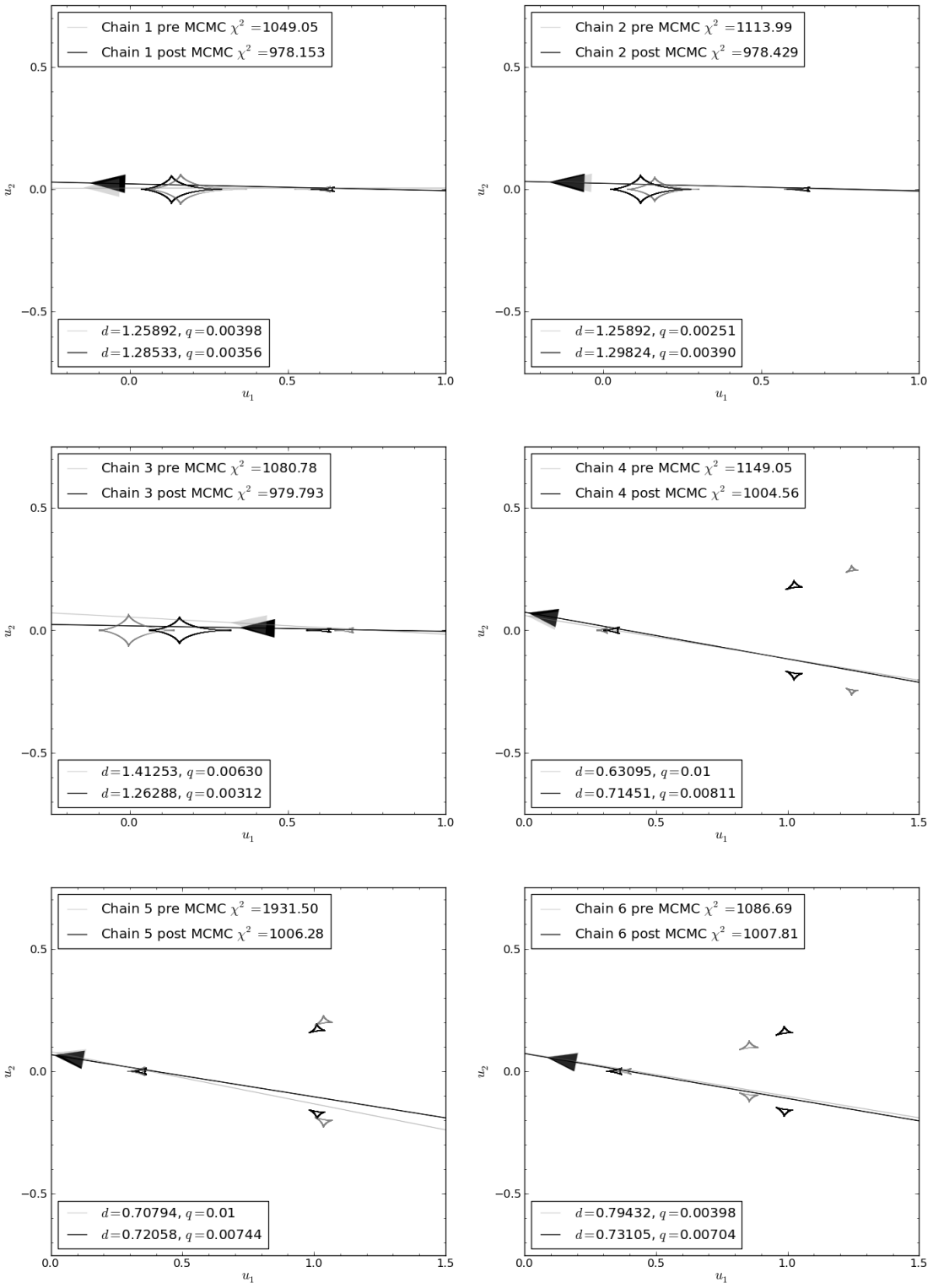


FIGURE 8.2: Caustic and trajectory evolution from the local MCMC search (Table 8.1) of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

TABLE 8.2: Seven parameter binary lens model solution from a search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241, determined using the GPU modelling method.

Parameter	GPU solution
d	1.295 ± 0.037
q	0.004 ± 0.013
ρ	$30.2 \pm 3.6 \times 10^{-4}$
u_0	$-43.4 \pm 5.9 \times 10^{-4}$
ϕ	3.109 ± 0.043
t_0	6730.747 ± 0.022
t_E	23.7 ± 1.4

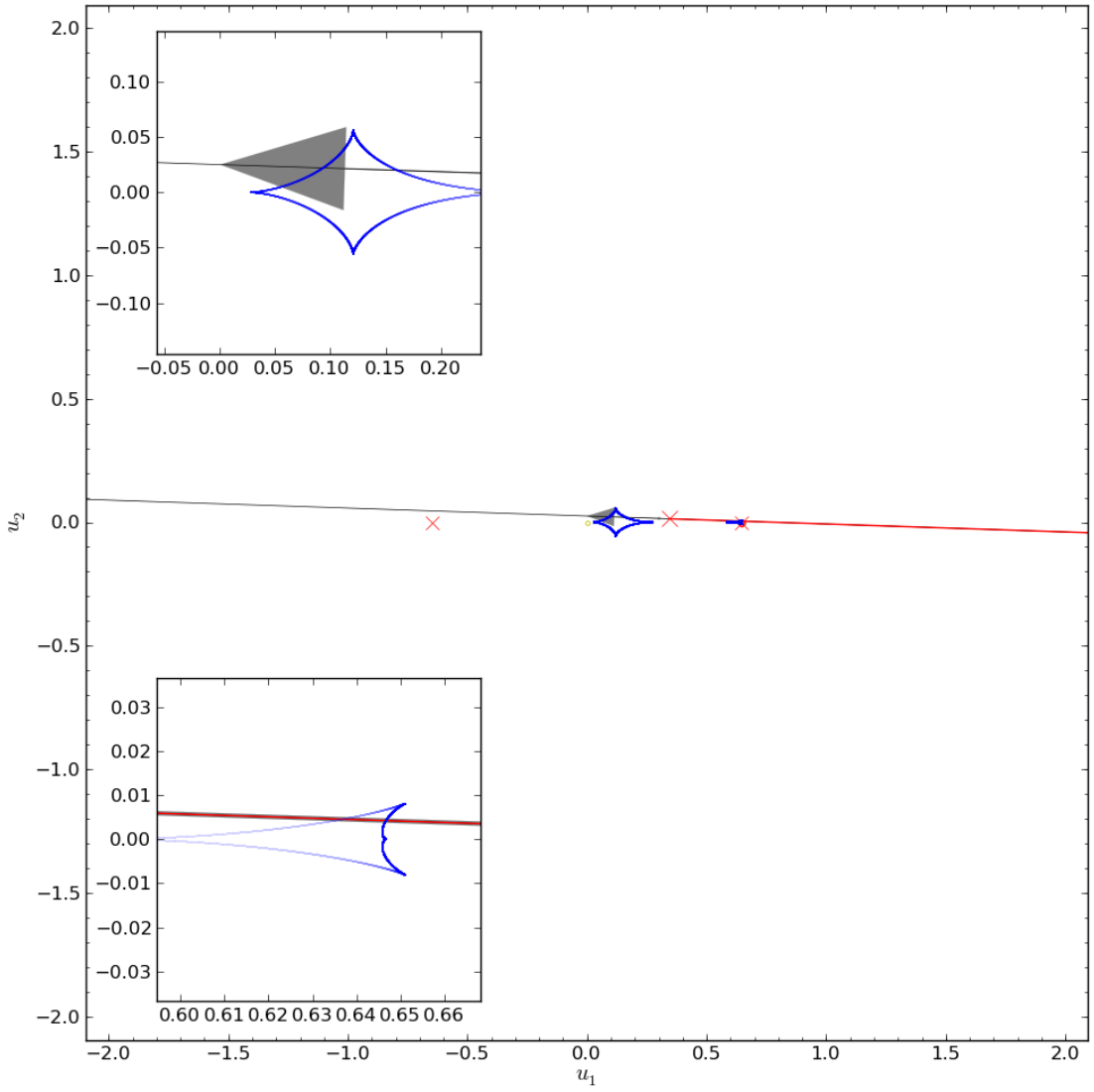


FIGURE 8.3: A caustic map with a overlaid trajectory for the model solution (Table 8.2) determined by the EMCEE GPU search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

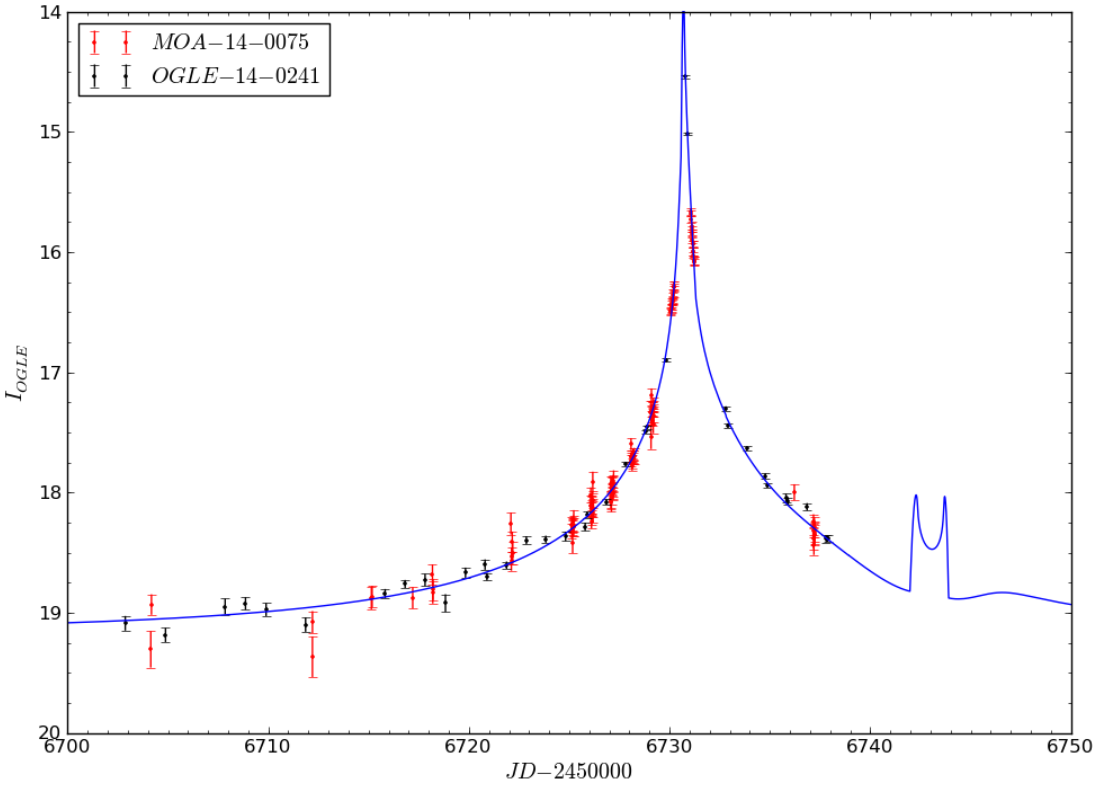


FIGURE 8.4: A light curve of the model solution (Table 8.2) determined by the GPU EMCEE search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

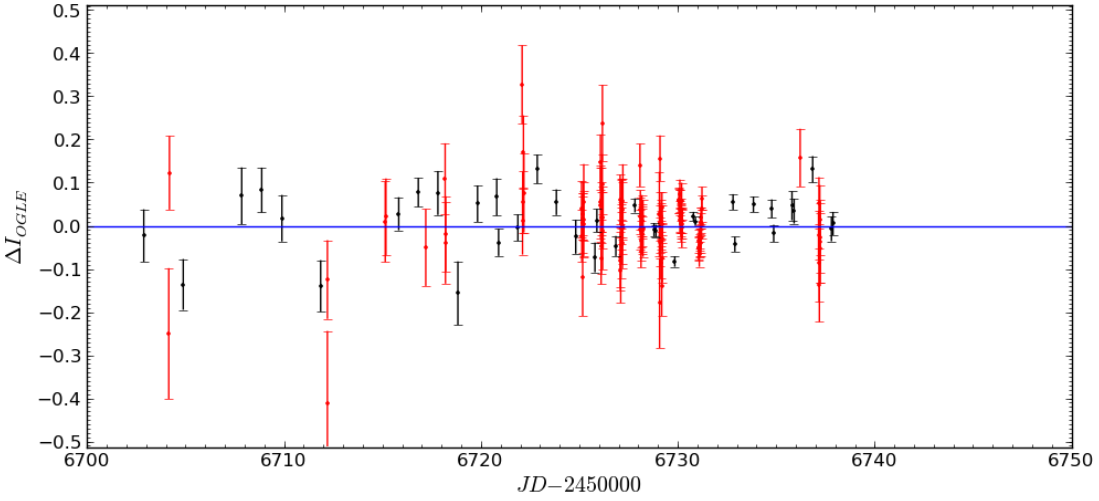


FIGURE 8.5: Light curve residuals of the model solution (Table 8.2) determined by the GPU EMCEE search of a partial data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

Later, on 2014 March 21 (AJD 6738.42) a public announcement of our planetary solution was made to the microlensing community, indicating the predicted caustic crossing, requesting for clarity of the two separate OGLE data sets, and for any additional observations if they exist. This was the first real time detection of a planet by our system, and shortly after two other modelling groups announced their agreement with our model presented here (C. Han, e-mail to ulens-analysis mail list on 2014 March 22 AJD 6738.80, T. Sumi, e-mail to ulens-analysis mail list on 2014 March 22 AJD 6738.85).

Seven days later (AJD 6743.37), the data showed no deviations in the falling shoulder as initially predicted, so re-modelling was initiated with the latest 256 data points from both OGLE and MOA. The modeller used the latest 256 data points from MOA (MOA-2014-BLG-0075) and OGLE (OGLE-2014-BLG-0241) as of AJD 6743.37 and was required to find a solution with a minimum change in magnitude of 4.25, and at least one peak located around AJD 6730.0.The solution showed that the predicted caustic crossing may have occurred, but could have fallen in a gap when no observations were made (AJD 6740 - 6742), meaning the planetary solution is still viable. This discovery was publicly announced to the microlensing community on 2014 March 28 (AJD 6743.58), and agreed upon via private communications. Due to the deviation occurring where no data exists it is difficult to confirm the parameter solution, and requests were made to re-reduce the data to help improve the model (C. Han, e-mail to ulens-analysis mail list on 2014 April 8 AJD 6755.59).

Once enough data had been collected to cover the whole event a re-modelling was performed (AJD 6819.35) to confirm the announced solution, and show agreement with the latest data. Using the latest 512 data points from each data set (MOA-2014-BLG-0075 and OGLE-2014-BLG-0241) as of AJD 6800.0 and the same model limitations as previously. This search lead to the minimized χ^2 map and markov chains shown in Table 8.3, and Figures 8.6 and 8.7. The best chain solution was chosen for a further EMCEE analysis using error normalised data, which determined the global solution shown in Table 8.4. This soution produces the caustic trajectory, light curve and residuals shown in Figures 8.8-8.10.

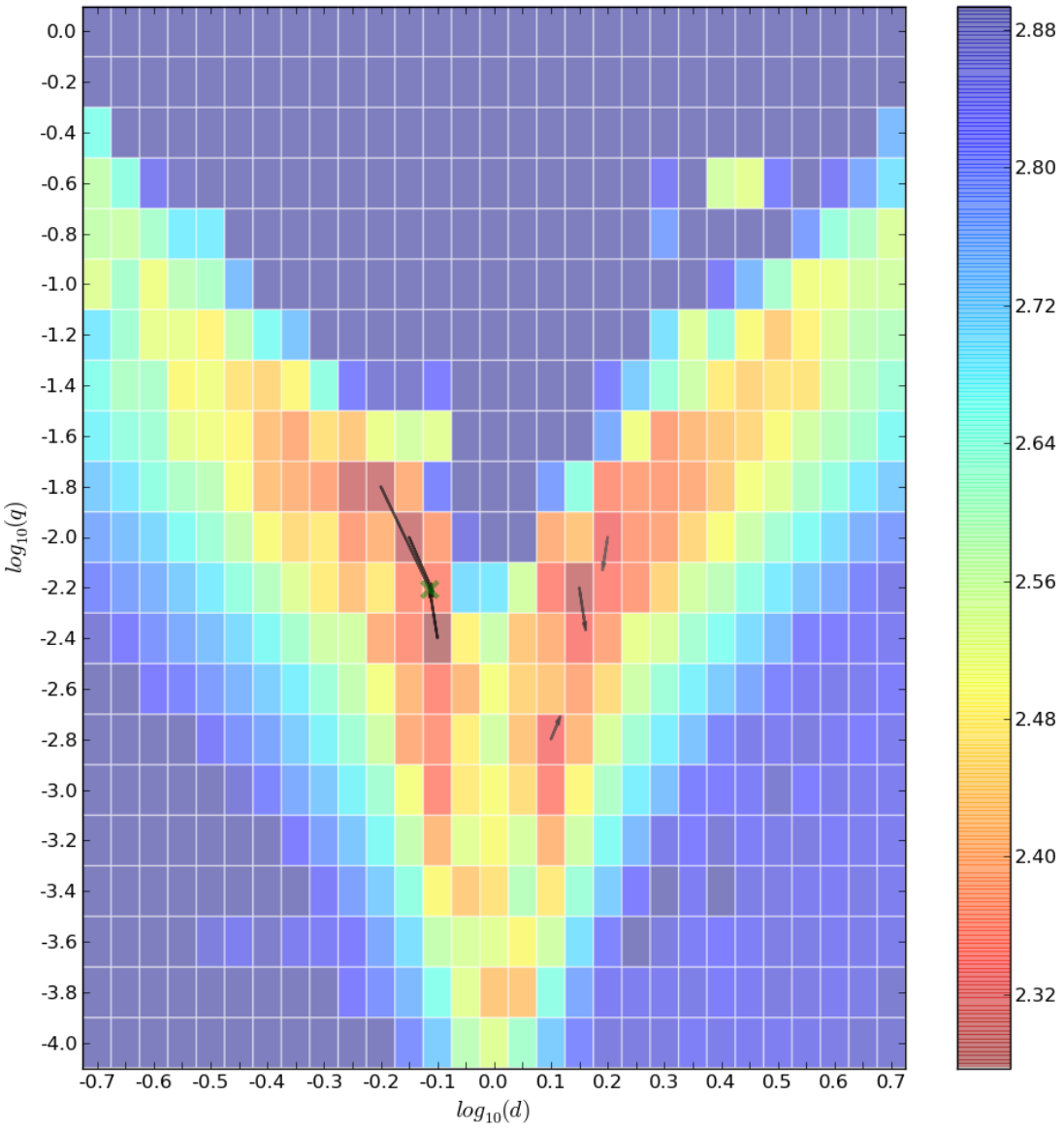


FIGURE 8.6: Local minima MCMC search (Table 8.3), showing d, q movement overlaid onto the initial minimized χ^2 d, q grid search map of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

TABLE 8.3: Seven parameter binary lens local minima model solutions from a search of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	0.76955	0.77905	0.78047	1.44912	1.30621	1.55241
q	0.00621	0.00611	0.00585	0.00430	0.00194	0.00742
ρ	0.00609	0.00609	0.00627	0.00532	0.00427	0.00515
u_0	−0.00974	−0.00935	−0.00914	−0.00632	−0.00523	−0.00580
ϕ	2.92833	2.92616	0.21237	3.10520	3.12020	3.06423
t_0	6730.7534	6730.741	6730.741	6730.863	6730.8141	6730.942
t_E	22.035	22.406	−22.344	25.693	29.742	28.037
χ^2	2285.434	2290.482	2290.662	2319.550	2323.823	2330.449

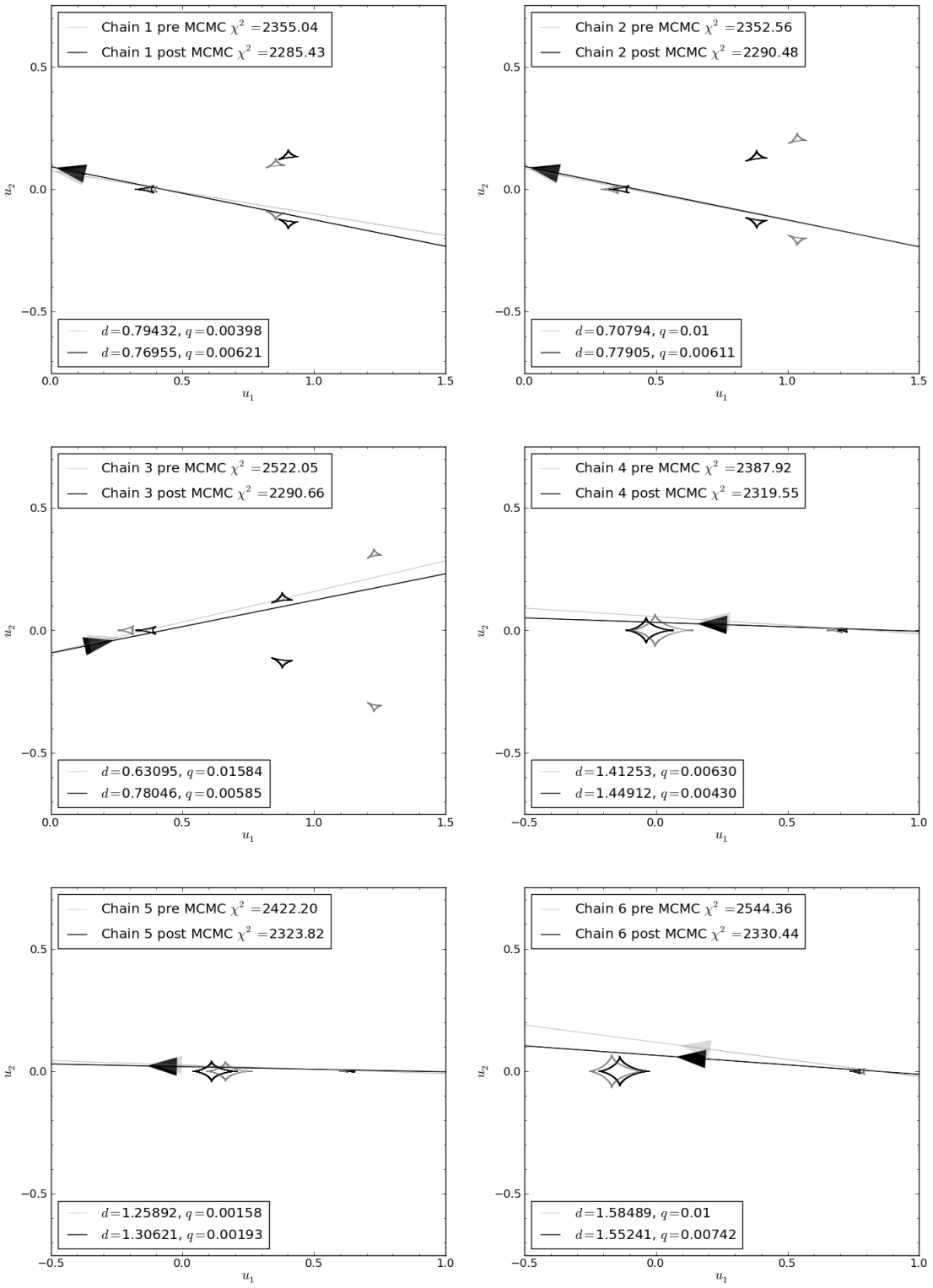


FIGURE 8.7: Caustic and trajectory evolution from the local MCMC search (Table 8.3) of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

TABLE 8.4: Seven parameter binary lens model solution from a search of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241, determined using the GPU modelling method.

Parameter	GPU solution
d	0.769 ± 0.011
q	$72.1 \pm 8.9 \times 10^{-4}$
ρ	$62.3 \pm 4.9 \times 10^{-4}$
u_0	$-107.4 \pm 9.5 \times 10^{-4}$
ϕ	2.911 ± 0.022
t_0	6730.76 ± 0.12
t_E	21.4 ± 1.0

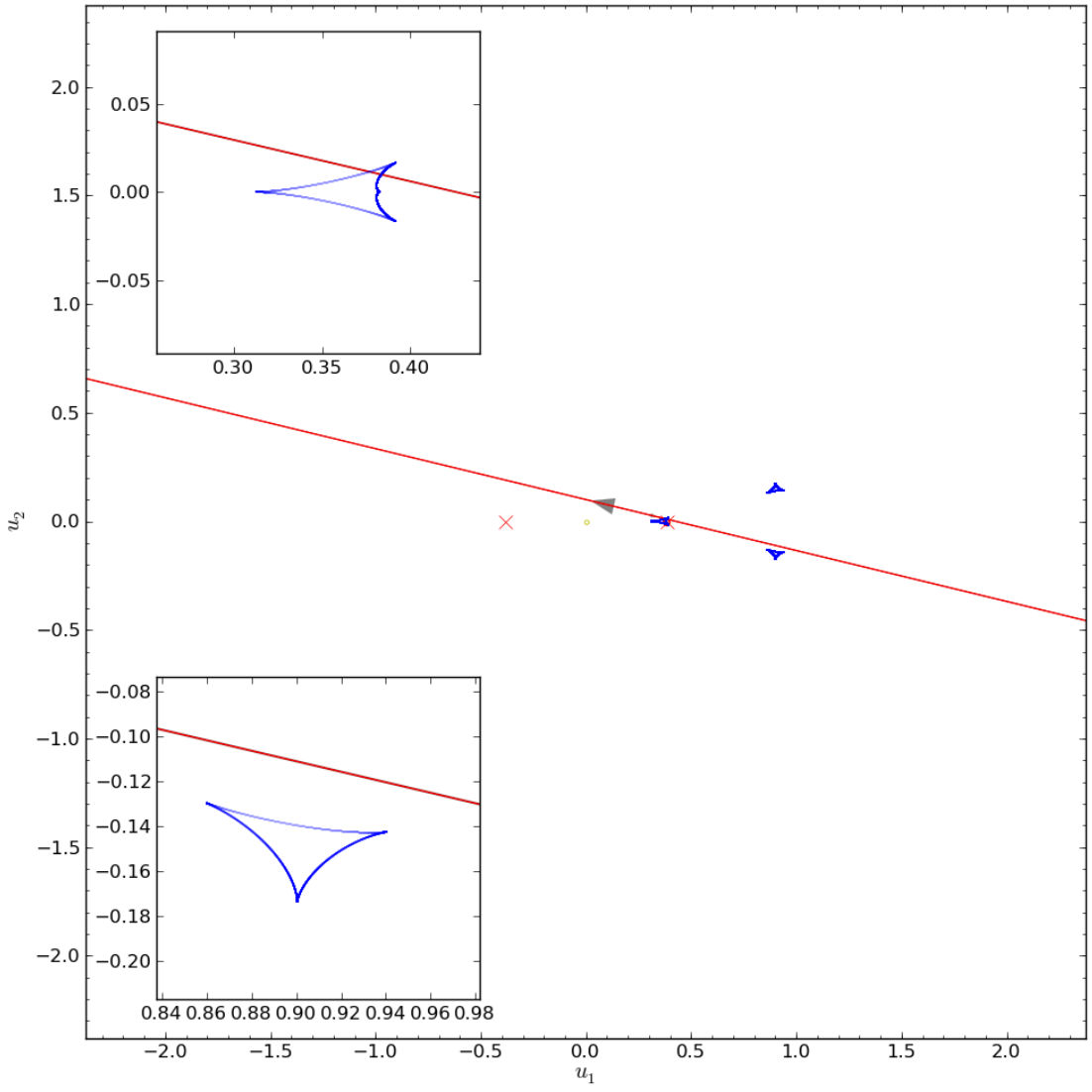


FIGURE 8.8: A caustic map with an overlaid trajectory for the model solution (Table 8.4) determined by the GPU EMCEE search of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

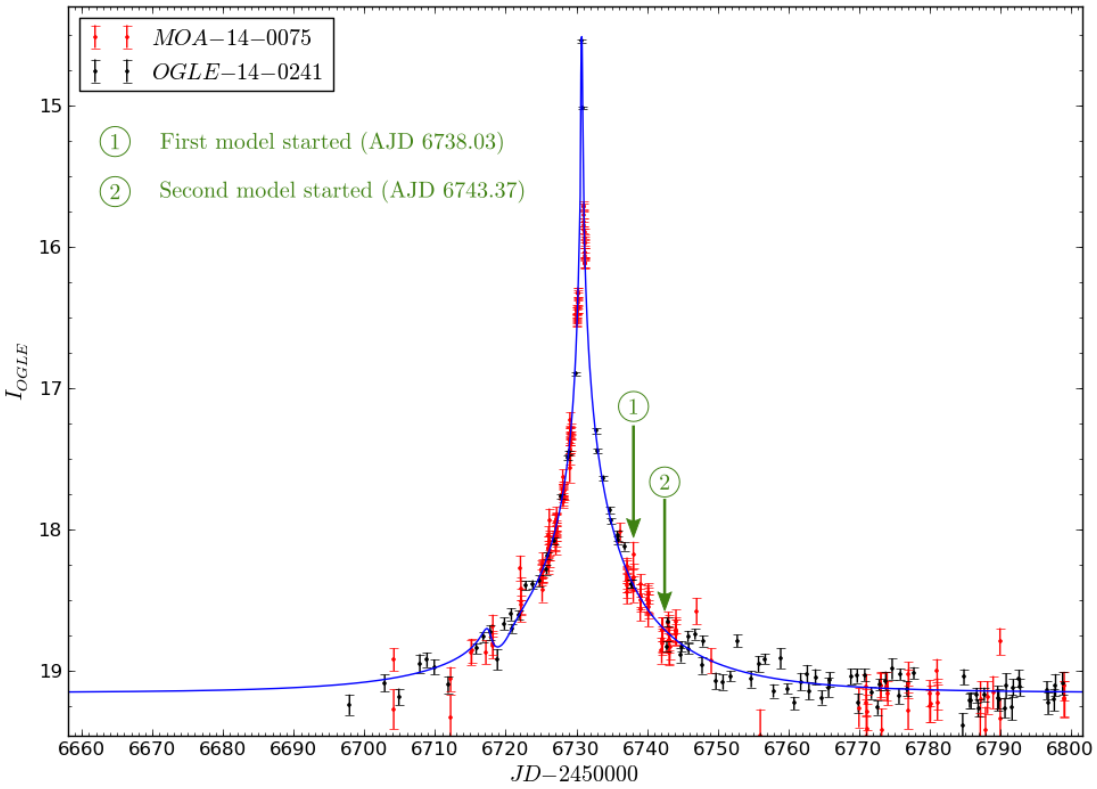


FIGURE 8.9: A light curve of the model solution (Table 8.4) determined by the GPU EMCEE search of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

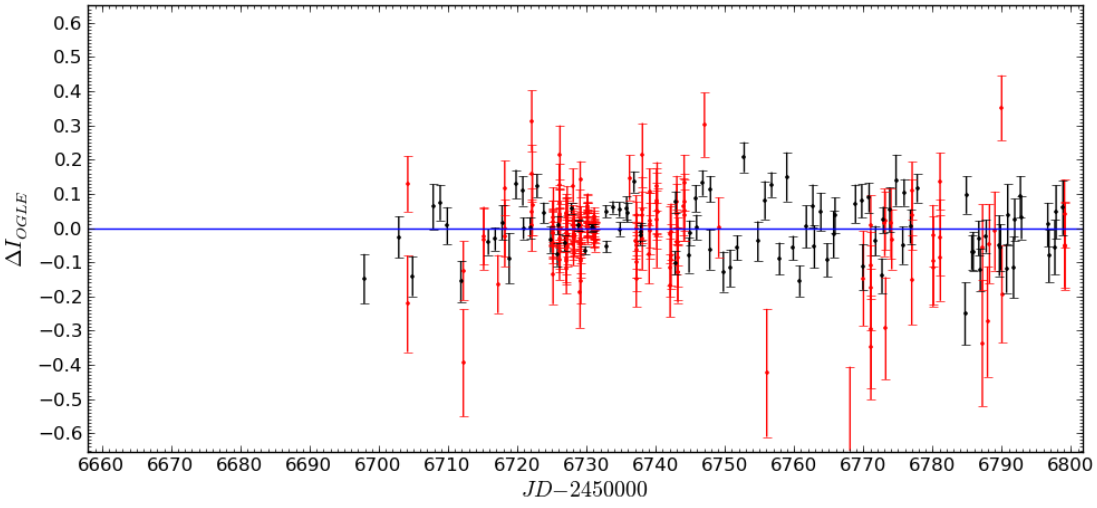


FIGURE 8.10: Light curve residuals of the model solution (Table 8.4) determined by the GPU EMCEE search of a full data set of MOA-2014-BLG-0075/OGLE-2014-BLG-0241.

The minimized χ^2 map of Figure 8.6 shows a similar shape to Figure 8.1, but has smaller regions of local minimum. These regions are explored further with local area MCMC searches and again show convergence for a close solution, but a greater diversity of wide solutions. Contrary to the previous minimization of the local chains with a partial data set, the close solution is now determined to be better, however, only by a $\Delta\chi^2 \approx 34$. This small χ^2 difference between the close and wide solutions is a good example of the close wide degeneracy experienced in binary lens models, which in this instance could have been resolved by higher cadence observations that could identify any possible additional features in the falling shoulder of the light curve. Although the separation parameter (d) cannot be very well constrained, both close and wide solutions agree upon the mass ratio being in the planetary range.

8.3 MOA-2014-BLG-153/OGLE-2014-BLG-0621

On 2014 April 30 (AJD 6778.19) an alert was sent out to the microlensing community that the event MOA-2014-BLG-153/OGLE-2014-BLG-0621 was starting to rise in magnification again, after passing the peak of what appeared to be an ordinary short duration single lens event (D. Bennett, e-mail to ulens-analysis mail list 2014 April 30 AJD 6778.19). However, the event had previously been spotted as being anomalous and our modeller had already been initialized on 2014 April 26 (AJD 6773.64). This first modelling was performed near the start of the event, when only a small quantity of data that described the anomaly existed (modelled with 294 MOA and 76 OGLE data points in the AJD range of 6600.00-6773.25). With the latest public announcement of the event being anomalous, more data existed, including the rise in magnification after passing through a trough. This new information provided a much better constraint on the model, prompting the modeller to be reinitialized. On 2014 May 2 (AJD 6779.55) the modeller was provided with 327 MOA and 83 OGLE data points between the AJDs of 6675.0 and 6779.6, and it was required to fit a model that would contain at least two peaks, with one being located near 6768.5, and achieve a minimum change in magnitude of 2.5 or more. The grid search and local area MCMC searches took a total of 2 hours and 40 minutes to complete, producing the results of Table 8.5 and Figures 8.11 and 8.12.

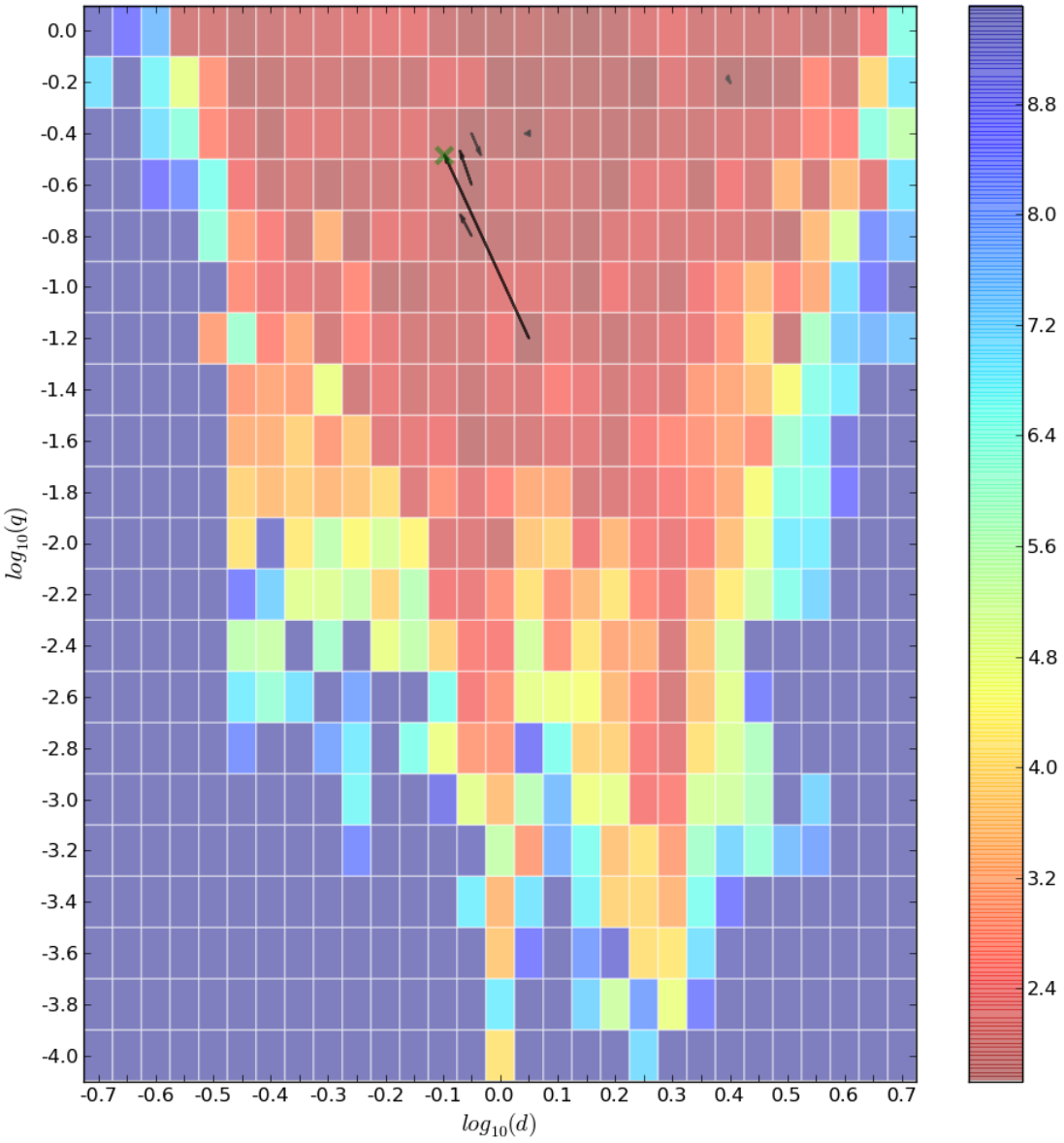


FIGURE 8.11: Local minima MCMC search of a full data set search movement overlaid onto the initial minimized χ^2 d, q map of MOA-2014-BLG-153/OGLE-2014-BLG-0621.

TABLE 8.5: Seven parameter binary lens local minima model solutions from a search of a full data set of MOA-2014-BLG-153/OGLE-2014-BLG-0621, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	0.80010	0.85139	0.85310	0.92444	1.10675	2.51561
q	0.32699	0.33870	0.19089	0.32887	0.39739	0.62681
ρ	4.47×10^{-4}	2.77×10^{-5}	3.41×10^{-4}	7.22×10^{-4}	1.42×10^{-5}	1.40×10^{-5}
u_0	-0.01029	0.01285	-0.01435	-0.01853	0.02776	-0.04247
ϕ	6.25477	6.31542	6.24472	6.24201	0.04442	6.25363
t_0	6777.343	6777.374	6778.546	6777.398	6777.255	6829.745
t_E	29.301	26.276	29.586	22.713	17.052	44.100
χ^2	669.310	670.803	673.948	675.399	686.564	692.671

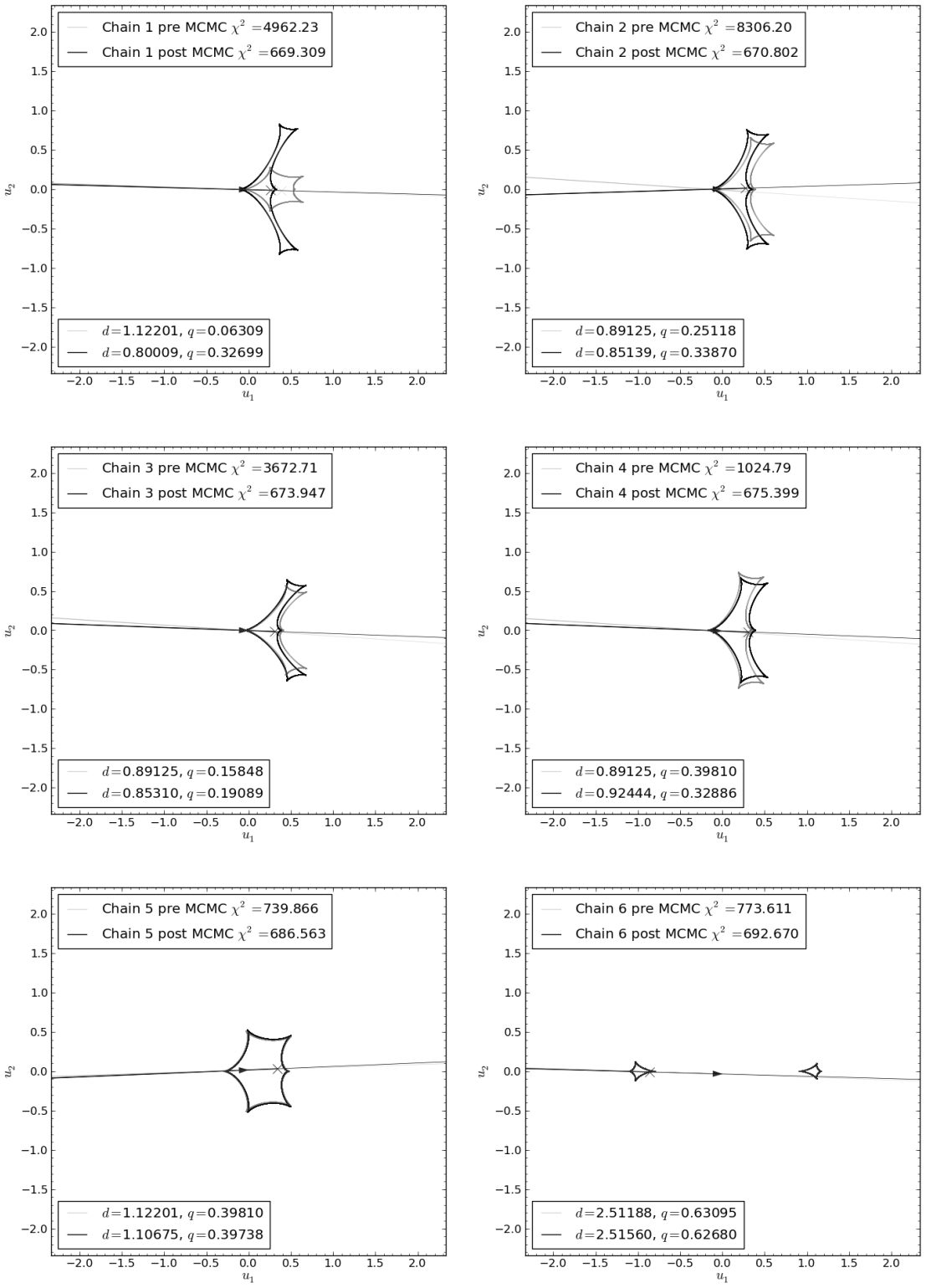


FIGURE 8.12: Local minima MCMC search of a full data set, chain evolution of caustic structures and trajectories (Table 8.5).

Of these solutions the lowest χ^2 chain was taken to be in the global minimum and used to start an EMCEE search to fine tune the solution and determine error estimates for each parameter. The result of this search is presented in Table 8.6 which produces the caustic with overlaid trajectory, light curve and residuals of Figures 8.13-8.15.

TABLE 8.6: Seven parameter binary lens model solution from a search of a full data set of MOA-2014-BLG-153/OGLE-2014-BLG-0621, determined using the GPU modelling method.

Parameter	GPU solution
d	0.799 ± 0.020
q	0.327 ± 0.026
ρ	$4.51 \pm 0.31 \times 10^{-4}$
u_0	-0.01027 ± 0.00091
ϕ	6.2504 ± 0.0061
t_0	6777.33 ± 0.24
t_E	29.20 ± 0.95

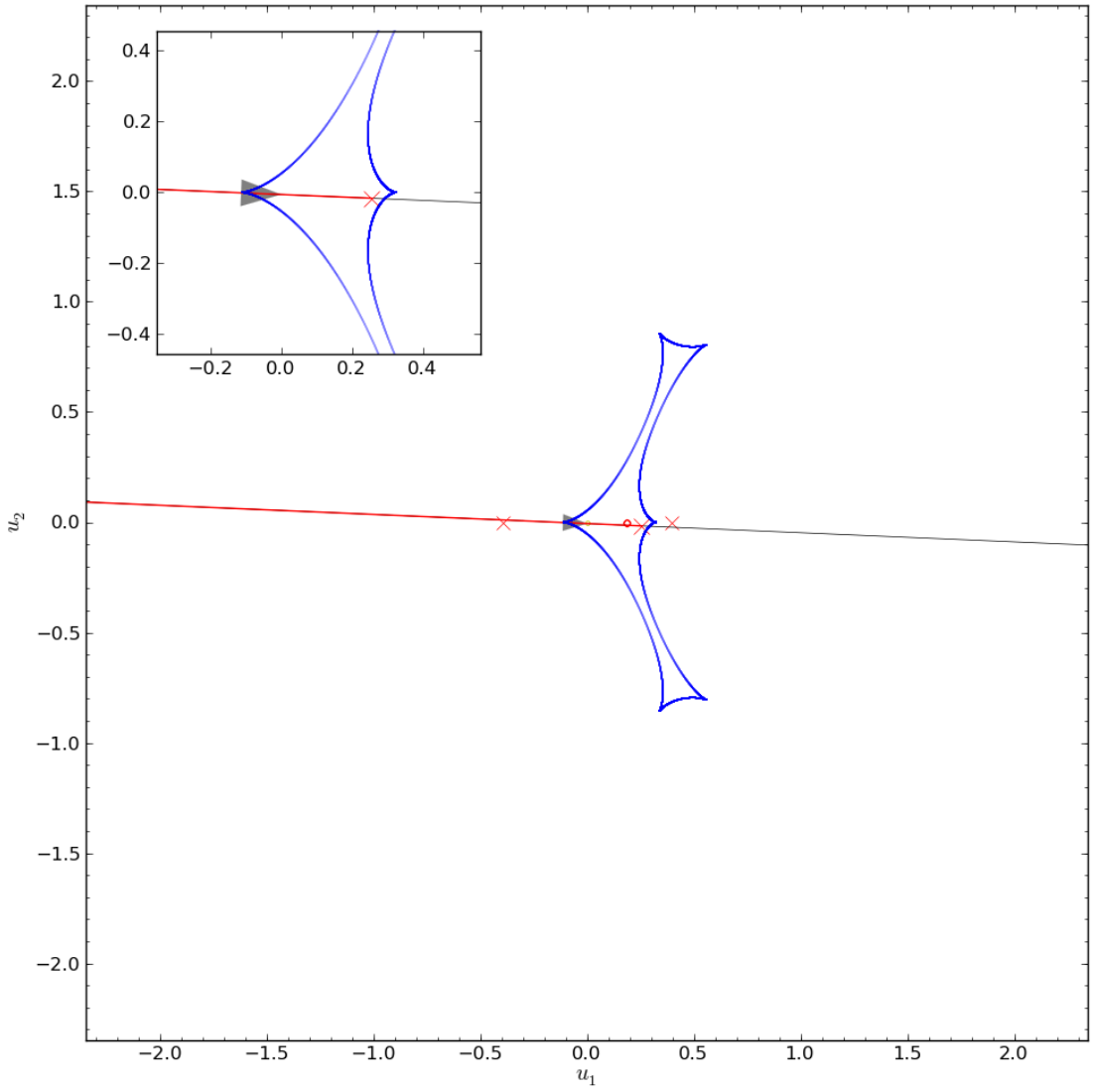


FIGURE 8.13: A caustic map with a overlaid trajectory for the model solution (Table 8.6) determined by the EMCEE GPU search of a full data set of MOA-2014-BLG-153/OGLE-2014-BLG-0621.

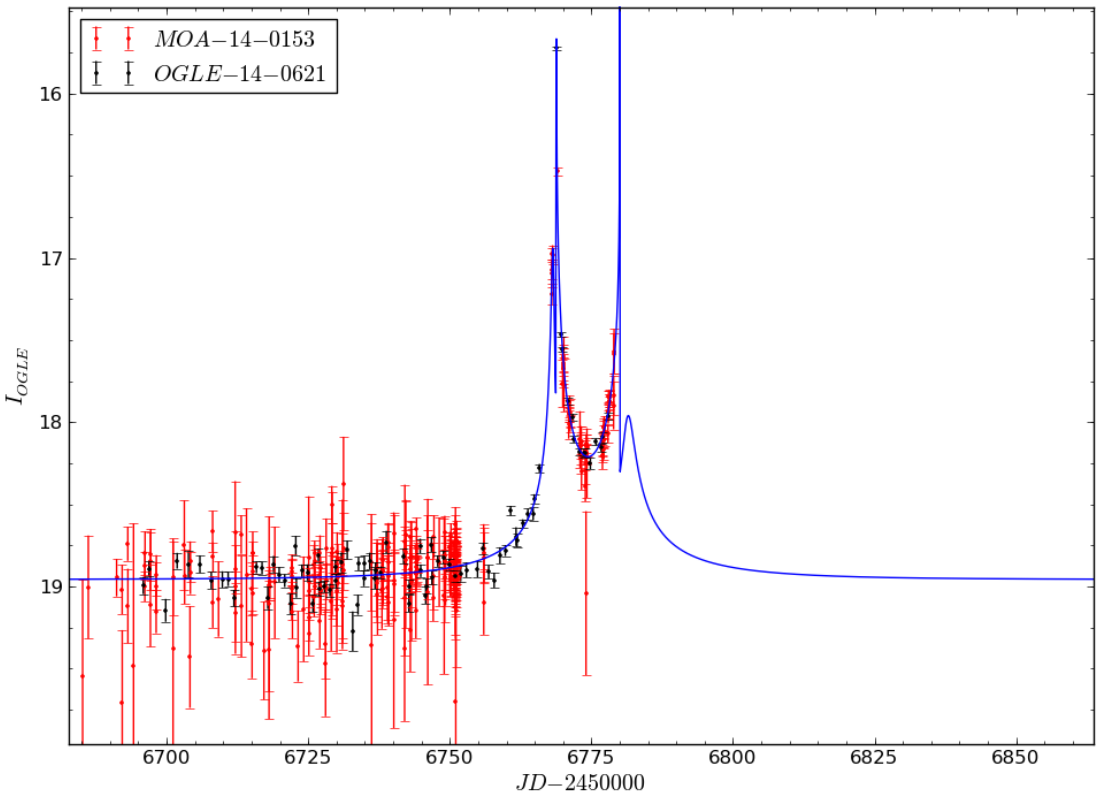


FIGURE 8.14: A light curve of the model solution (Table 8.6) determined by the GPU EMCEE search method on a full data set of MOA-2014-BLG-153/OGLE-2014-BLG-0621.

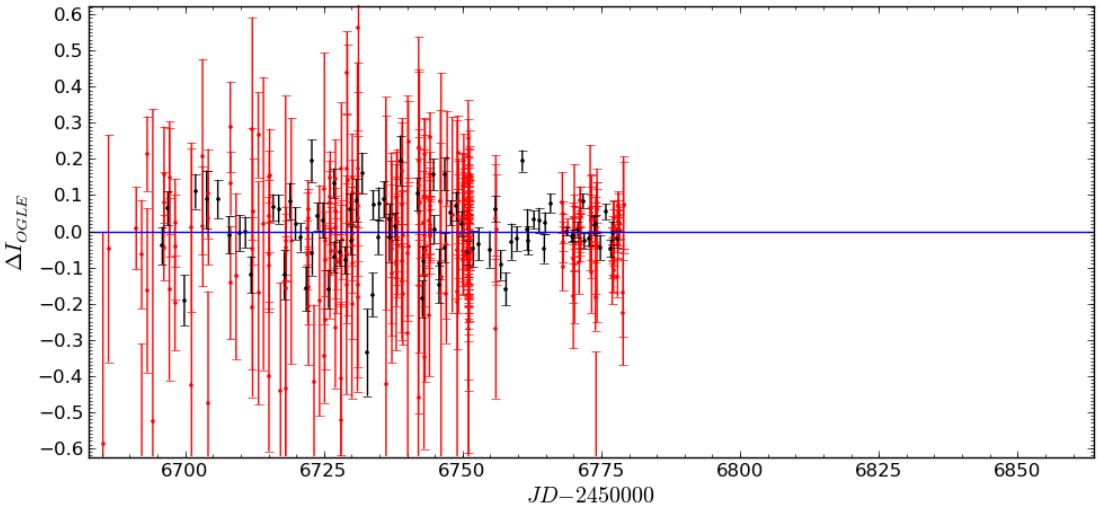


FIGURE 8.15: Light curve residuals of the model solution (Table 8.6) determined by the GPU EMCEE search method on a full data set of MOA-2014-BLG-153/OGLE-2014-BLG-0621.

Table 8.5 and Figure 8.12 show that the first four solutions presented all result in very similar solutions with comparable χ^2 values. This solution was also identified in the previous modelling performed on 2014 April 26. However, without the additional constraint in the location of the trough between the caustic crossings, other competitive models existed. Of the local area MCMC searches previously performed, solutions similar to those of Chains 5 and 6 in Table 8.5 were found, with no indication of a preferential solution.

The solution presented in Table 8.6 was publicly announced to the microlensing community on 2014 May 2 (AJD 6773.82). The announcement showed the above model with the predicted peak after the caustic crossing and stated that the model is not well constrained, due to data not yet covering the full event. Nine days later, on 2014 May 11 (AJD 6788.20), data existed that showed the caustic exit had passed and the magnification was returning to baseline. With this latest data another modelling group announced their results showing they had found the same solution as we presented (T. Sumi, e-mail to ulens-analysis mail list 2014 May 11 AJD 6788.70), and the following day an additional modelling group also confirmed our result (C. Han, e-mail to ulens-analysis mail list 2014 May 12 AJD 6789.60). To re-confirm the previously determined models, on 2014 May 23 (AJD 6800.67) and 2014 July 15 (AJD 6861.50), re-modelling of this event was performed on the latest available data. In both cases the same global minimum area was located showing the same light curve shape and caustic structure. The more recent model, which included baseline data after the event, produced a slightly larger separation and lower mass ratio parameter, coming from the extra data that added a constraint on when the event returned to baseline.

This event shows the success of our modeling procedure correctly identifying early on, a valid model solution of an on-going binary lens event. It was confirmed by multiple modelling groups, and later re-confirmed by our own modelling procedure when additional data that covered the full duration of the event was included.

8.4 MOA-2014-BLG-0205/OGLE-2014-BLG-0626

On 2014 May 15 (AJD 6793.08) an e-mail alert was posted by Gould to the microlensing community about an anomalous rise on the falling shoulder of the event designated MOA-2014-BLG-0205/OGLE-2014-BLG-0626 (Gould, e-mail to ulens-analysis mail list on 2014 May 15 AJD 6793.08). Due to technical reasons the observational data that captured this rise in magnification was not made publicly available as early as usual, meaning modelling could be not initiated instantly. This issue was resolved the following day, and when it was made available additional data existed that showed the rise

in magnification up to the peak of the feature. Initial predictions from other modelers (D. Bennett, e-mail to ulens-analysis mail list on 2014 May 16 AJD 6794.43) believed the deviation to be caused by a planetary object and follow up observations were recommended. Due to the initial alert, follow up observations had been made from CTIO and the RoboNet group. These observations covered gaps in the MOA data, showing the early rise of the anomaly and more of the fall after it peaked. It was at this point that our modelling system was initiated.

Starting our modelling procedure on the 2014 May 16 (AJD 6793.98) with the latest 256 data points from MOA and OGLE, 31 from Robonet Sutherland Dome A, and 8 from Robonet Siding Spring Dome A. The model was required to contains two peaks located around 6790.0 and 6793.2 with a minimum magnification change of 1.0. The initial grid search for areas of local minima and MCMC search to explore them completed in 2 hours and 42 minutes producing the following minimized χ^2 map shown in Figure 8.16 and the six parameter solutions in Table 8.7, with their caustic structures shown in Figure 8.17.

TABLE 8.7: Seven parameter binary lens local minima model solutions of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	0.31794	0.33867	4.40606	0.49613	3.57392	1.95450
q	0.44481	0.74856	0.69170	0.11396	0.38499	0.07243
ρ	1.340×10^{-7}	0.00172	1.821×10^{-5}	0.01732	0.01366	0.00740
u_0	−0.06021	−0.07945	1.31983	−0.05316	0.68162	0.02531
ϕ	4.15768	2.60448	4.07296	4.02184	4.07793	3.93275
t_0	6791.969	6791.891	6836.197	6791.941	6813.594	6794.642
t_E	34.570	31.691	43.756	35.804	40.275	43.50238
χ^2	717.622	724.732	725.041	743.351	771.530	800.540

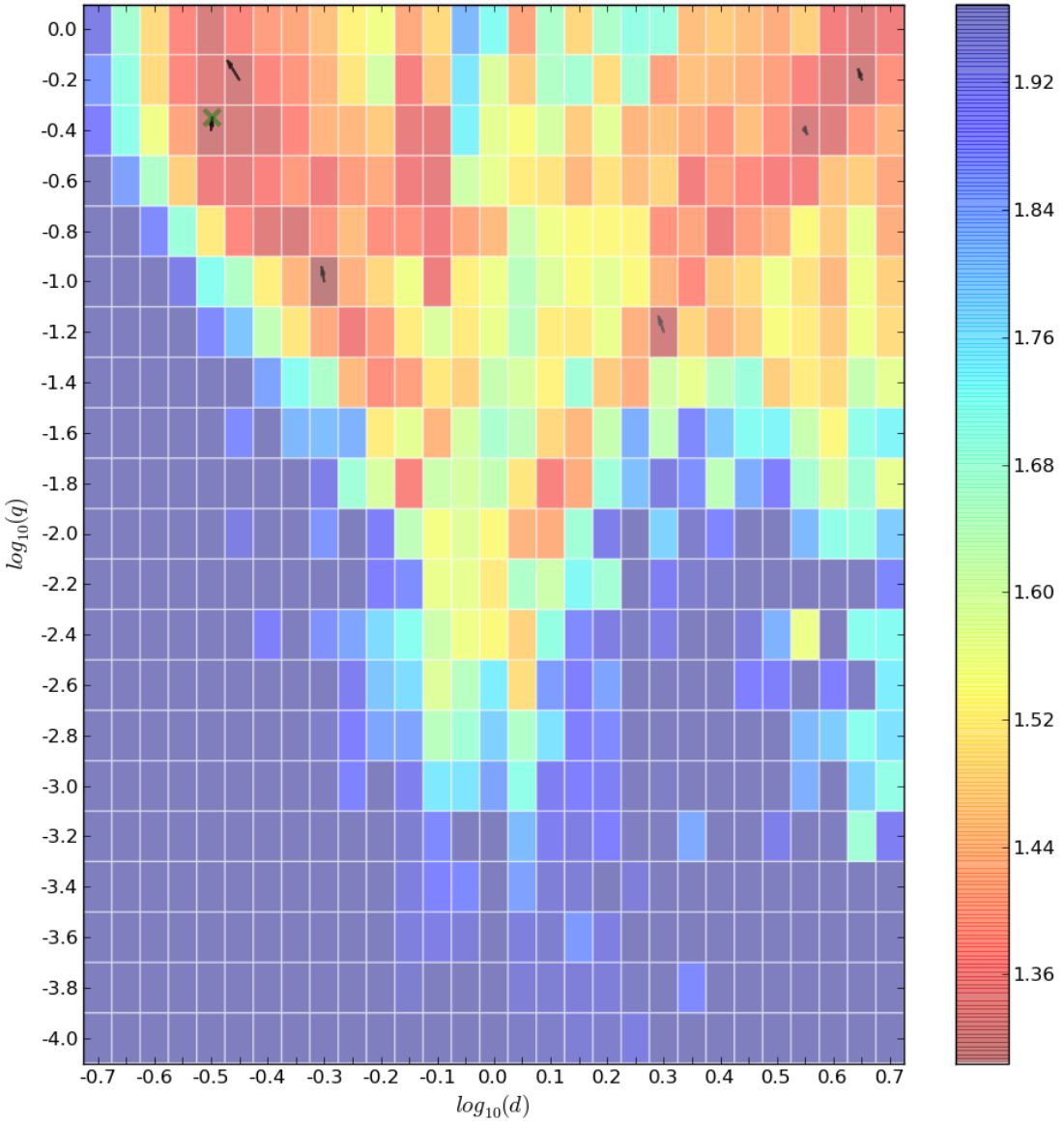


FIGURE 8.16: Local minima MCMC search movement overlaid onto the initial minimized χ^2 d, q map of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

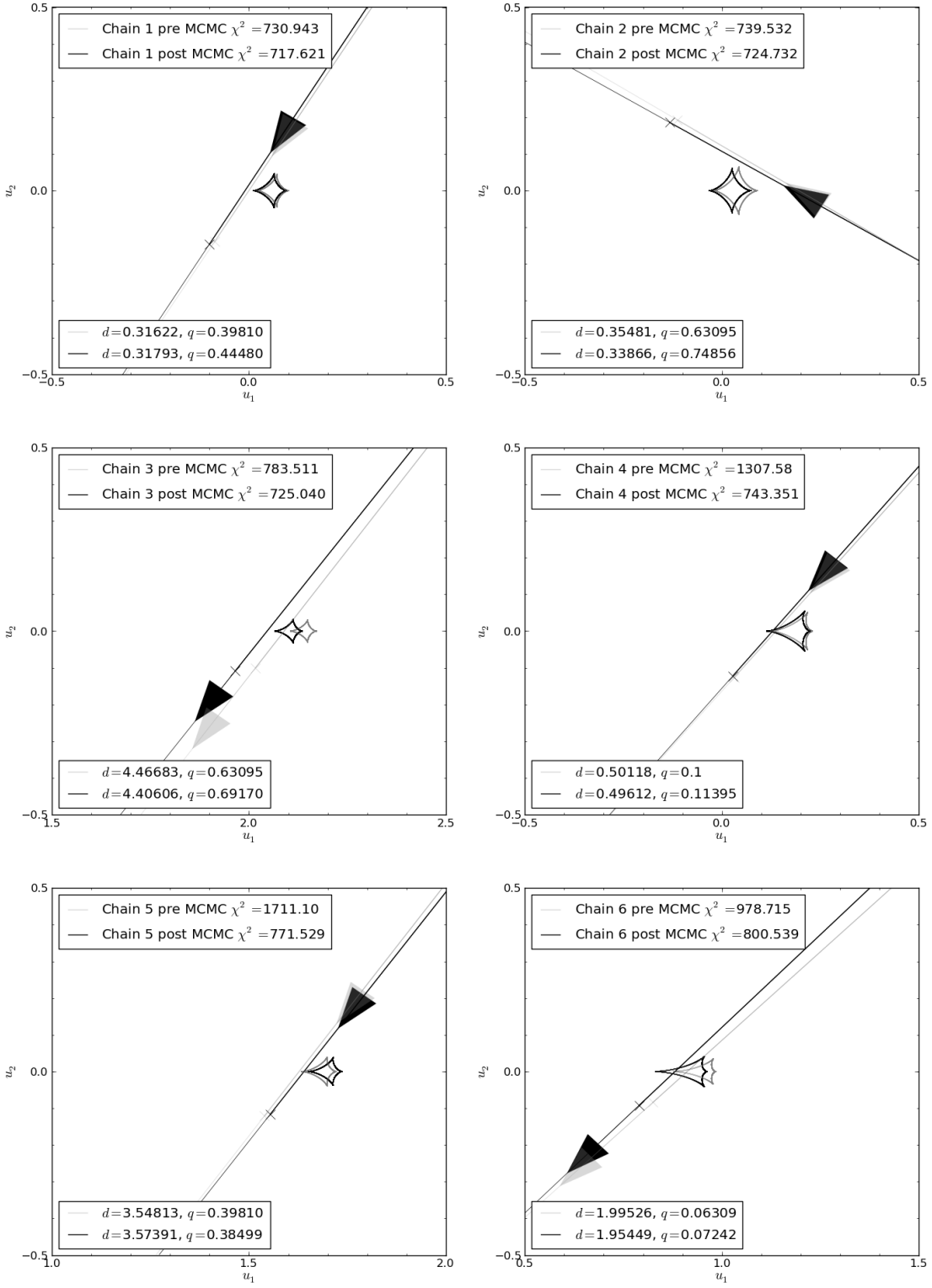


FIGURE 8.17: Local minima MCMC search of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626 showing the evolution of caustic structures and trajectories of the parameter sets (Table 8.7).

Of the six solutions the lowest χ^2 parameter set was processed further using an EMCEE search with error-normalized data, which located the global minimum with the parameters shown in Table 8.8. This solution produced the caustic trajectory, light curve and residuals shown in Figures 8.18-8.20.

TABLE 8.8: Seven parameter binary lens model solution from the search of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626, determined using the GPU modelling method.

Parameter	GPU solution
d	0.321 ± 0.010
q	0.406 ± 0.061
ρ	$11.9 \pm 4.1 \times 10^{-8}$
u_0	-0.0594 ± 0.0034
ϕ	4.153 ± 0.026
t_0	6791.97 ± 0.27
t_E	34.6 ± 1.8

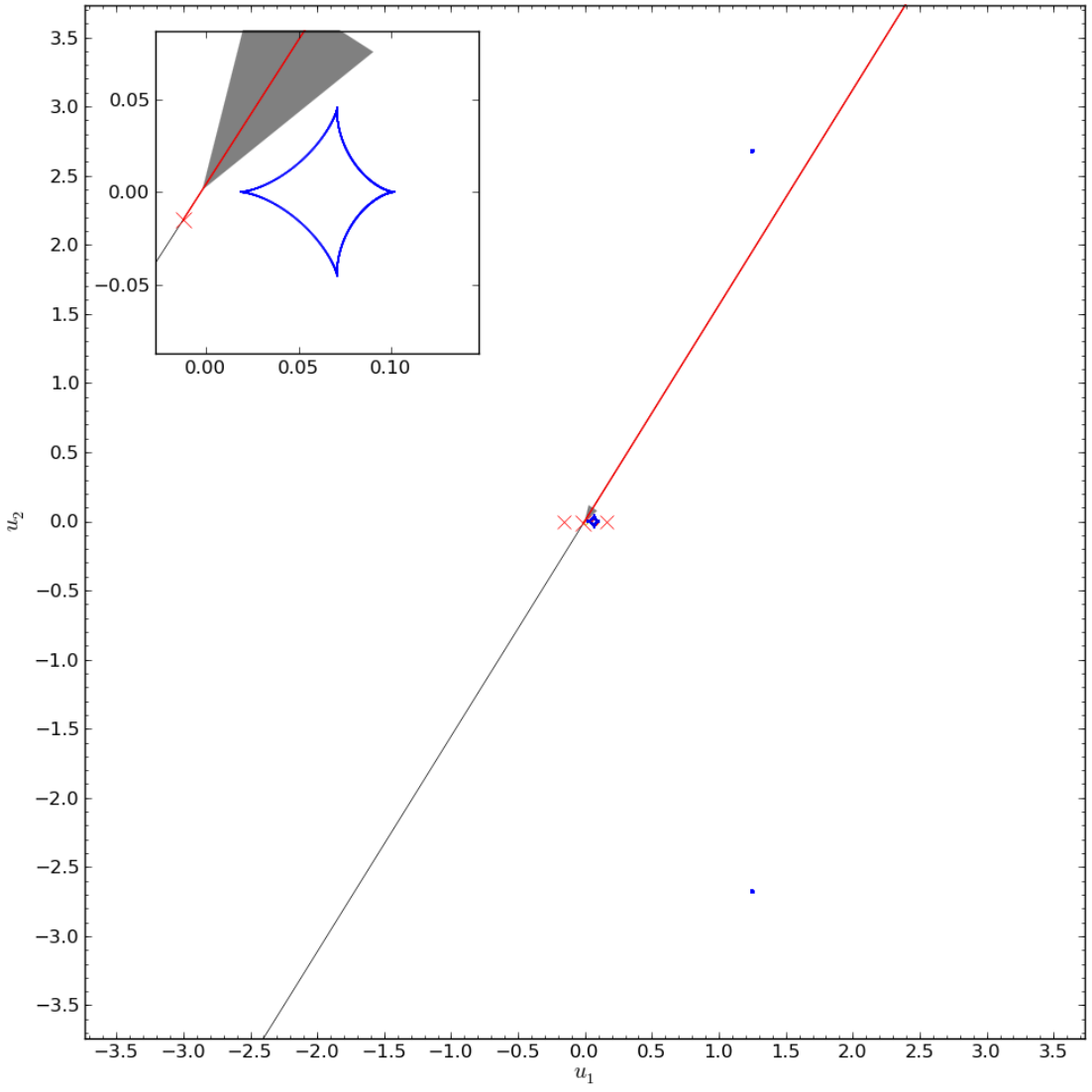


FIGURE 8.18: A caustic map with a overlaid trajectory for the model solution (Table 8.8) determined by the GPU EMCEE search method of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

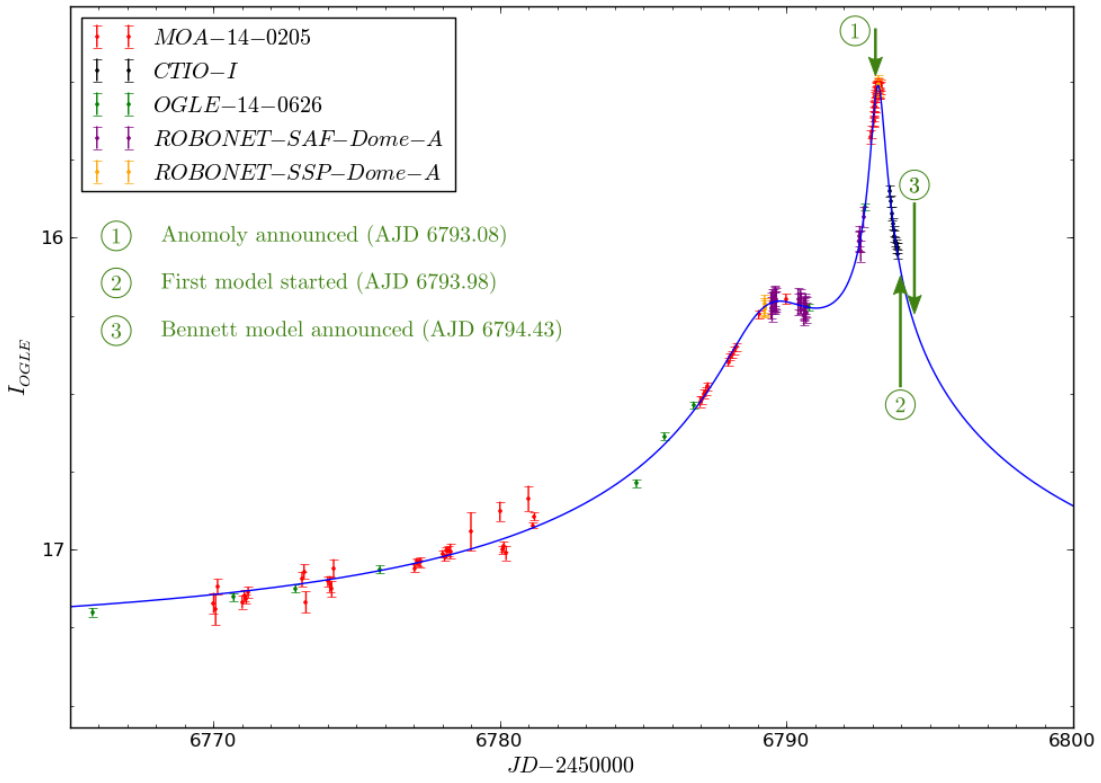


FIGURE 8.19: A light curve of the model solution (Table 8.8) determined by the GPU EMCEE search method of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

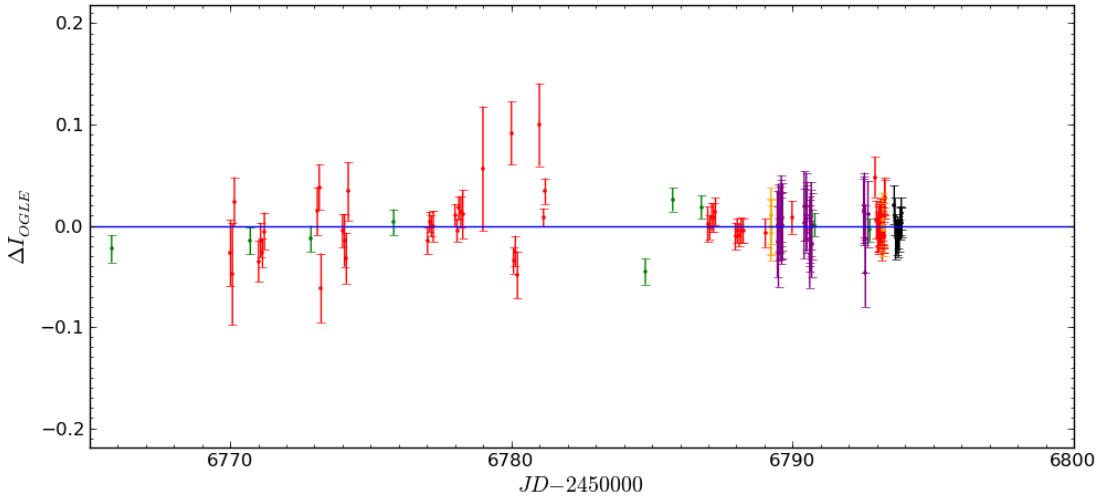


FIGURE 8.20: Light curve residuals of the model solution (Table 8.8) determined by the GPU EMCEE search method of a partial data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

Table 8.8 and Figures 8.18-8.20 show the predicted model for MOA-2014-BLG-0205/OGLE-2014-BLG-0626 to be a binary lens system with a mass ratio ≈ 0.4, the minimized χ^2 map (Figure 8.16) shows no competing planetary solutions as were initially suspected. From initializing the modelling process to determining where the global minimum and other local minimum are, the procedure took 2 hours and 42 minutes. A solution set was publicly announced to the microlensing community on the 2014 May 17 (AJD 6794.61), and over the following few days private communications from other modelling groups confirmed the binary lens solution presented here (V. Bozza, e-mail to ulens-analysis mail list on 2014 May 17 AJD 6795.54, and 19 AJD 6797.36, and C. Han, e-mail to ulens-analysis mail list on 2014 May 18 AJD 6796.68, and 2014 June 6 AJD 6814.75).

With the most recent data set as of 2014 June 9 (AJD 6817.69), re-modelling of this event (with 192 data points from MOA, 25 from CTIO I filter, 35 from OGLE, 31 from Robonet Sutherland Dome A, and 21 from Robonet Siding Spring Dome A) produced a similar minimized χ^2 map to before (Figure 8.16). The additional data provides greater constraints on the parameter solutions, resulting in the convergence of the local MCMC searchs shown in Figure 8.21. These chains produced the six solutions shown in Table 8.9 and Figure 8.22, showing a $\Delta\chi^2 = 0.294$ between the best two solutions, highlighting the degeneracy between close and wide solutions.

TABLE 8.9: Seven parameter binary lens local minima model solutions from a search of a full data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	4.01740	0.35687	0.36209	0.35553	4.19129	4.54388
q	0.68663	0.36929	0.35972	0.36123	1.24245	0.85379
ρ	0.00411	9.788×10^{-5}	0.00470	1.269×10^{-4}	0.00386	3.132×10^{-7}
u_0	1.15435	−0.07186	−0.07270	−0.07160	1.35752	1.83790
ϕ	4.05781	4.13004	4.12783	4.13169	0.93509	0.94683
t_0	6824.903	6791.881	6791.887	6791.893	6830.489	6848.529
t_E	35.722	29.059	28.702	28.934	37.152	41.777
χ^2	426.315	426.609	427.037	427.130	427.217	430.703

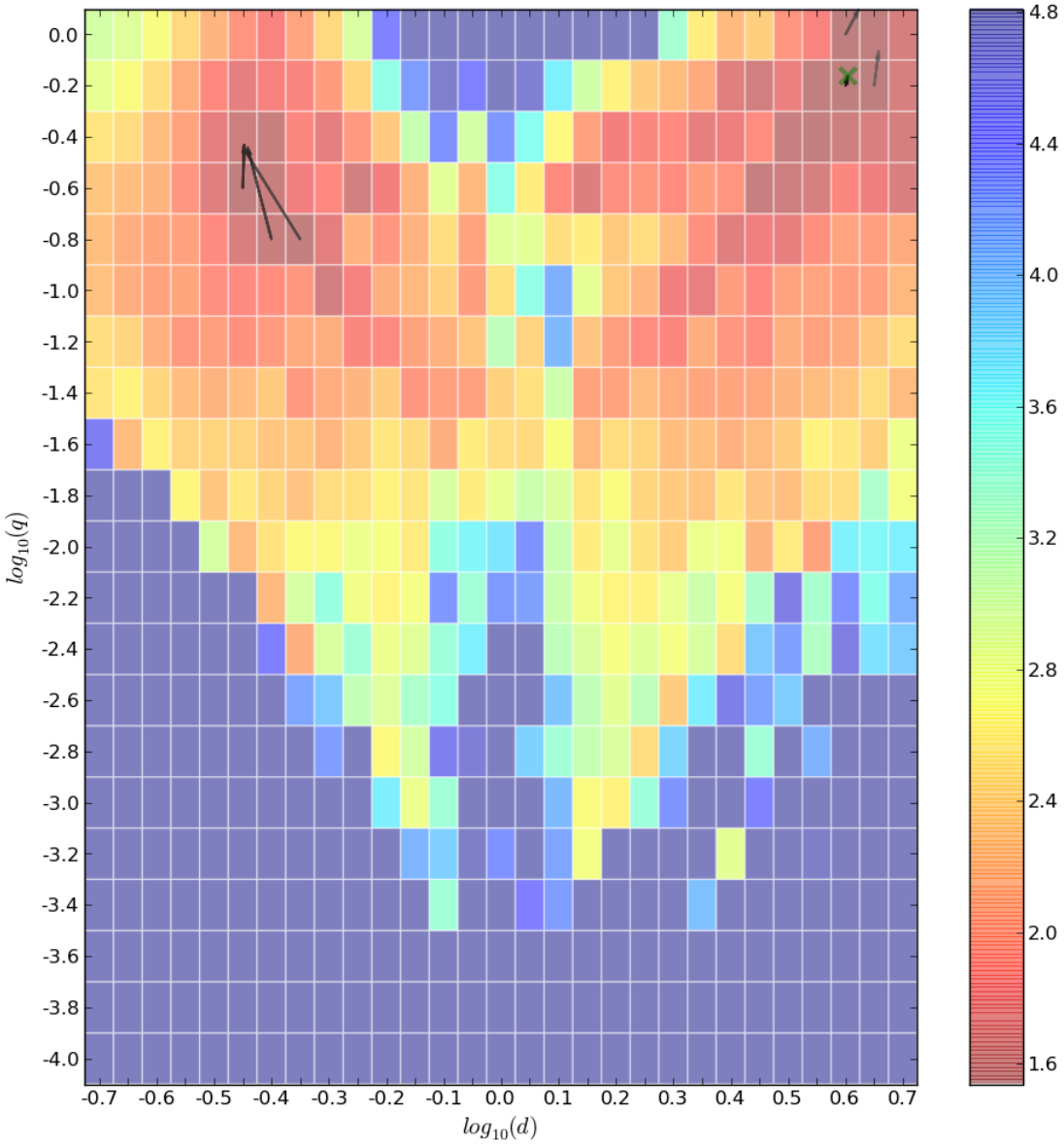


FIGURE 8.21: Local minima MCMC search of a full data set search movement overlaid onto the initial minimized χ^2 d, q map of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

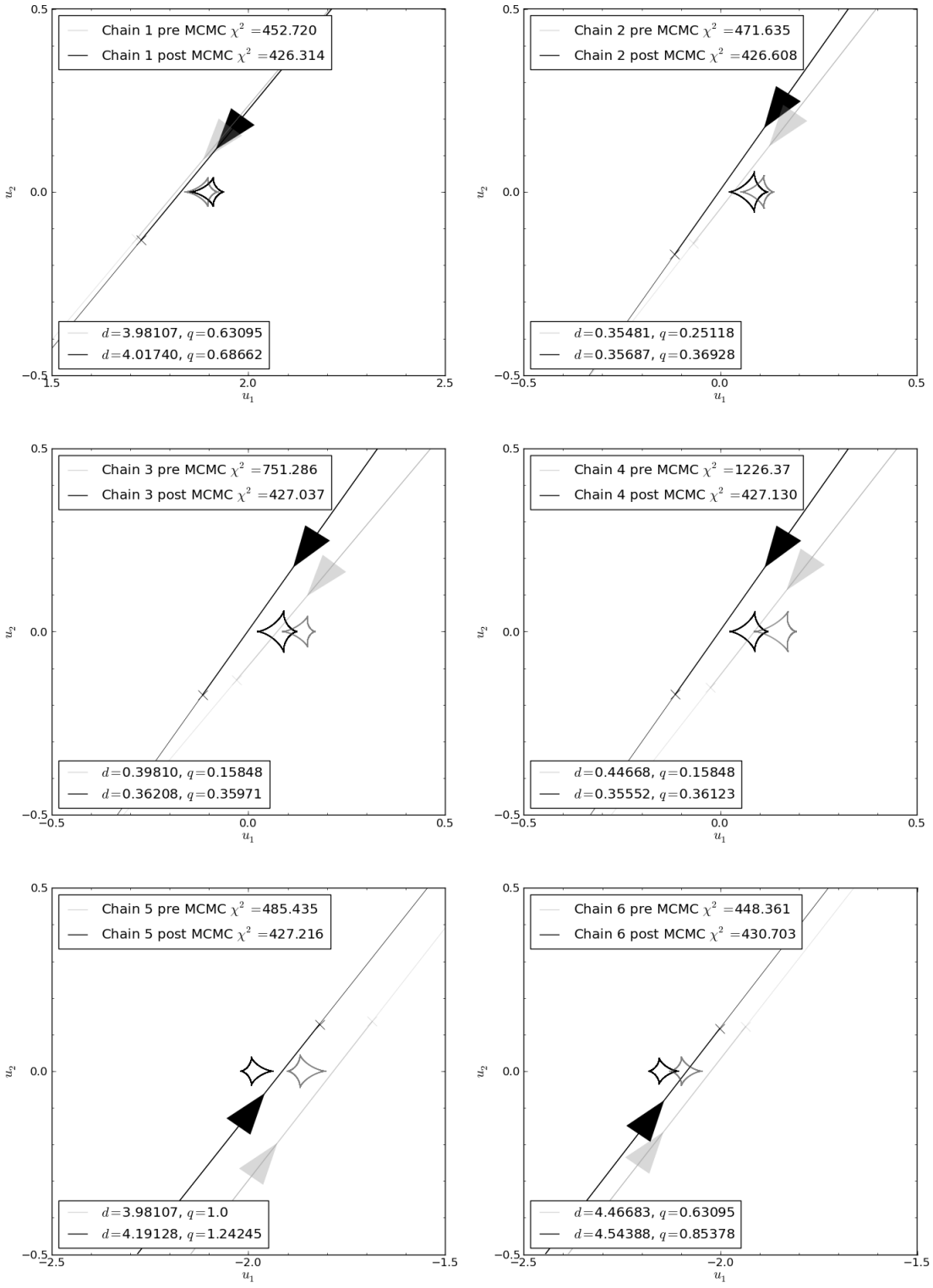


FIGURE 8.22: Local minima MCMC search of a full data set, chain evolution of caustic structures and trajectories (Table 8.9).

Choosing the best solution, in this case the wide solution, the EMCEE process was initiated with normalised data, leading to a new parameter solution set shown in Table 8.10 producing the following caustic trajectory, light curve and residuals shown in Figures 8.23-8.25.

TABLE 8.10: Seven parameter binary lens model solution from a search of a full data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626, determined using the GPU modelling method.

Parameter	GPU solution
d	4.049 ± 0.075
q	0.674 ± 0.024
ρ	0.00404 ± 0.00044
u_0	1.162 ± 0.031
ϕ	4.068 ± 0.026
t_0	6825.2 ± 1.2
t_E	36.6 ± 1.1

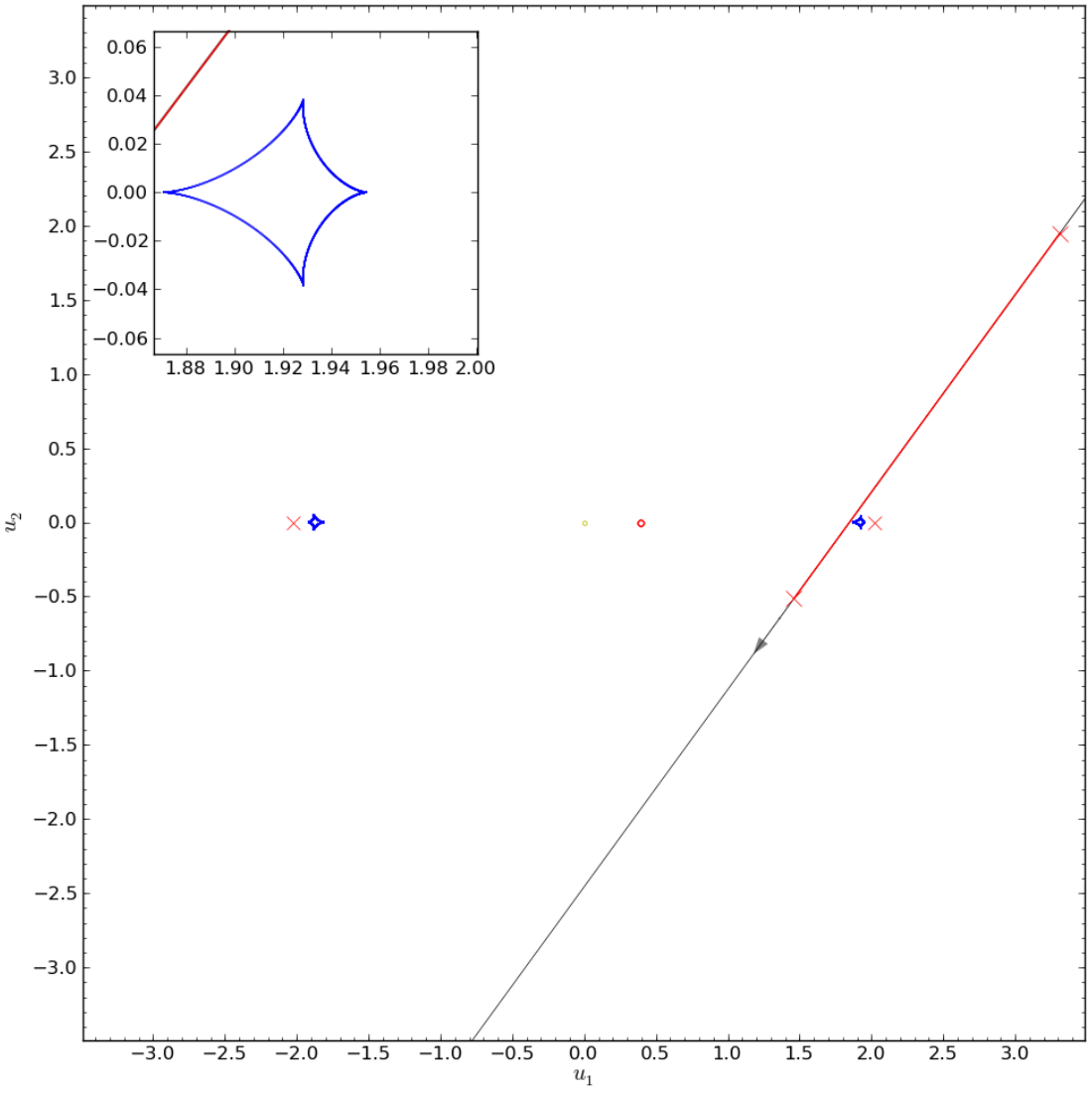


FIGURE 8.23: A caustic map with a overlaid trajectory for the model solution (Table 8.10) determined by the EMCEE GPU search of a full data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

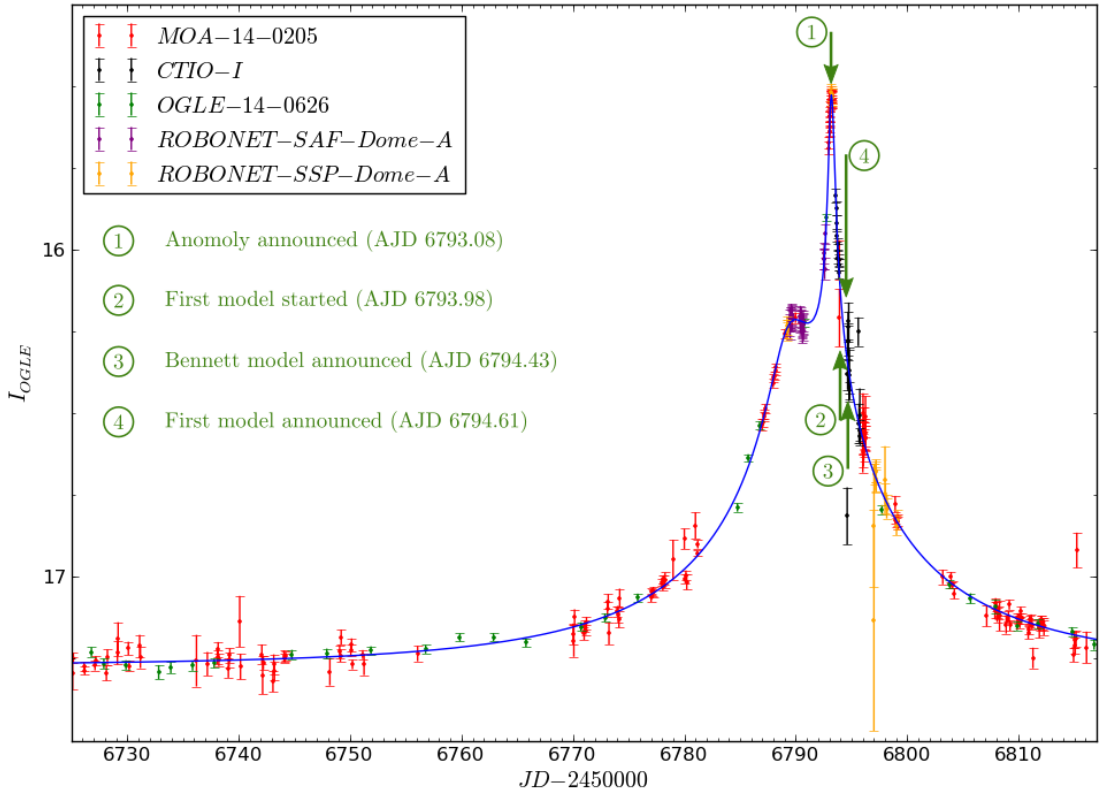


FIGURE 8.24: A light curve of the model solution (Table 8.10) determined by the GPU EMCEE search method on a full data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

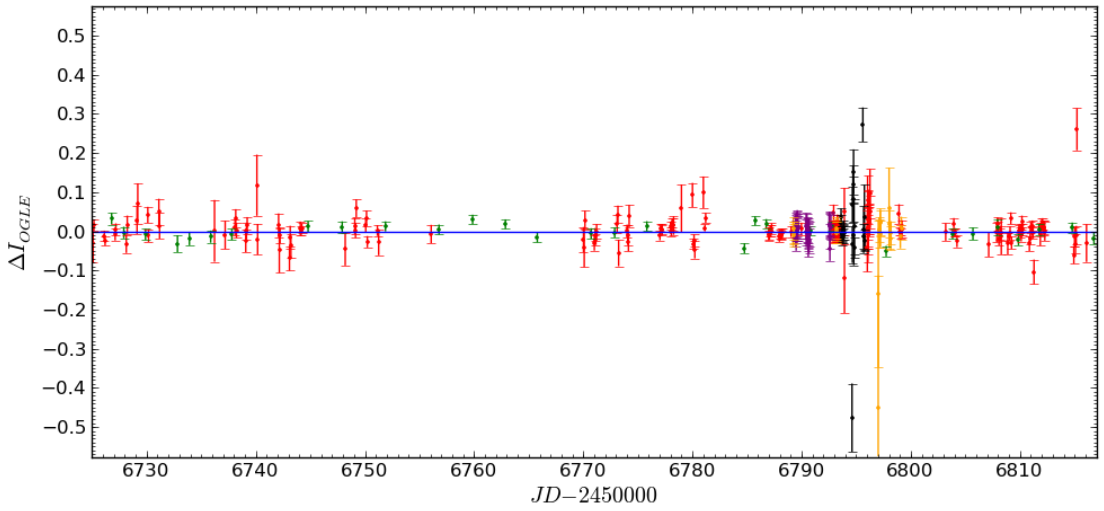


FIGURE 8.25: Light curve residuals of the model solution (Table 8.10) determined by the GPU EMCEE search method on a full data set of MOA-2014-BLG-0205/OGLE-2014-BLG-0626.

The analysis of this microlensing event shows the importance of a real-time binary lens modelling system such as the one presented here. It determined a viable solution in under three hours showing that planetary solutions were unlikely, country to initial suspicions. The predicted model was then later confirmed with a completed data set showing the initial predictions to be accurate.

8.5 OGLE-2014-BLG-0945

On 2014 May 31 (AJD 6808.52) a private communication was sent out to the microlensing community proposing a planetary model for the event OGLE-2014-BLG-0945 (V. Bozza, e-mail to ulens-analysis mail list on 2014 May 31 AJD 6808.52). The model was fitted soon after the main deviation from the single lens model occurred, meaning there was not a large quantity of data to help constrain the model. Together with the poorly sampled anomaly, these are the likely causes of the poorly constrained model shown in Table 8.11.

TABLE 8.11: Seven parameter binary lens model solution proposed by V. Bozza via private communications to the microlensing community of event OGLE-2014-BLG-0945.

Parameter	GPU solution
d	1.66397 ± 5.70855
q	0.0246901 ± 0.182593
ρ	0.0152152 ± 5.03105
u_0	1.15998 ± 18.4276
ϕ	1.81615 ± 4.93321
t_0	6803.68 ± 4.64239
t_E	7.17082 ± 69.6675

Nearly a month later (on 2014 July 24, AJD 6832.54), when more observations had been taken that showed the remainder of the event, our modeling procedure was initialized to investigate the validity of the previously proposed planetary model. The modelling procedure was started with the requirement to produce a model that had at least one peak located around the AJD 6806.0 and achieve a minimum change in magnitude of 2.0 or more. The model was provided with 63 OGLE data points in the AJD range 6500.0 to 6832.5, it took 2 hours and 5 minutes to complete the grid search and six local area MCMC searches. This search produced the results of Table 8.12, and Figures 8.26 and 8.27.

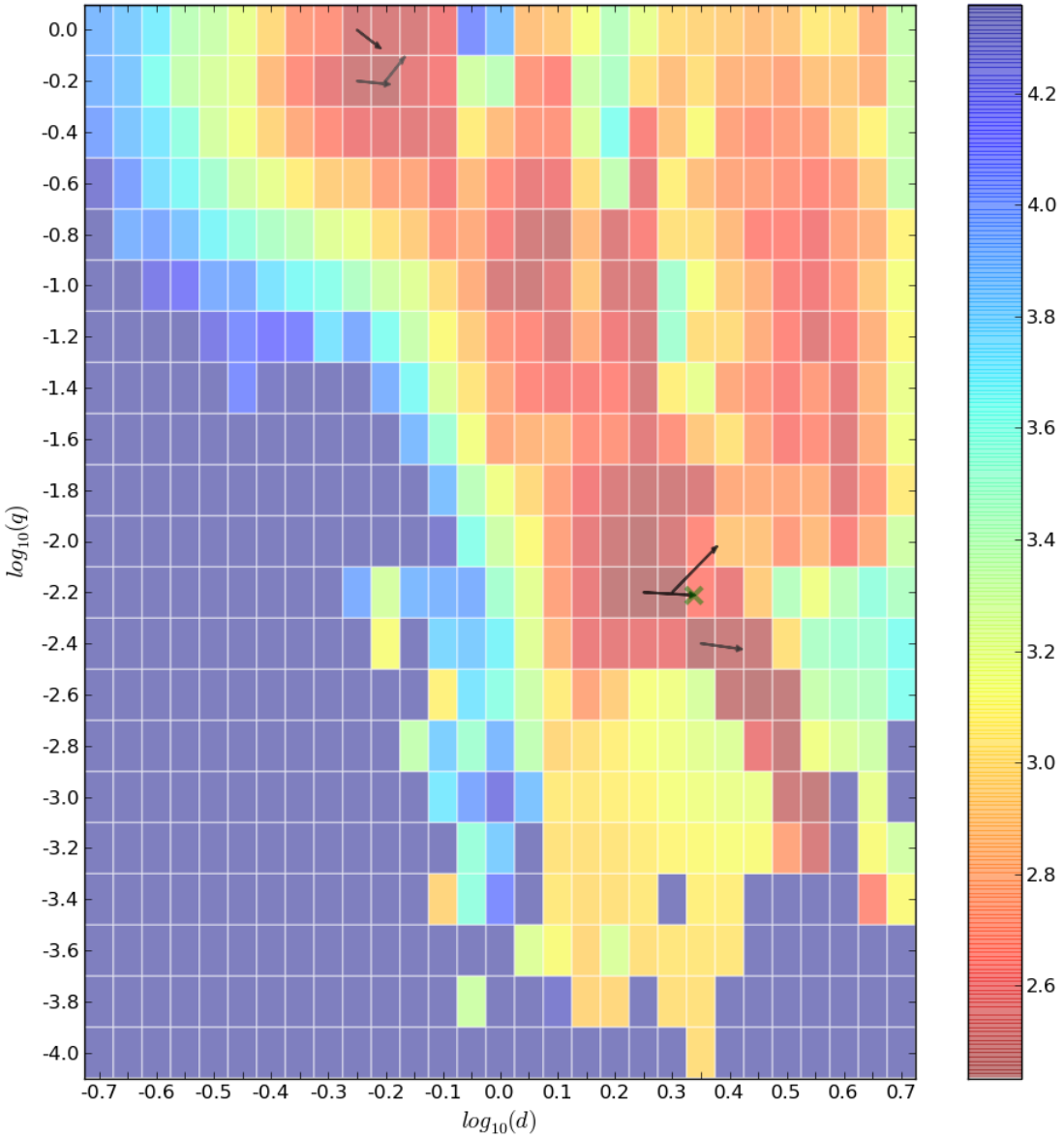


FIGURE 8.26: Local minima MCMC search of a full data set search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2014-BLG-0945.

TABLE 8.12: Seven parameter binary lens local minima model solutions from a search of a full data set of OGLE-2014-BLG-0945, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	2.18127	2.38523	0.61745	2.63212	0.63956	0.68058
q	0.00615	0.00951	0.84943	0.00378	0.61277	0.78144
ρ	0.11815	0.14968	0.00338	0.15661	0.00140	1.57×10^{-6}
u_0	−1.65915	1.82432	−0.14118	2.21973	−0.15656	−0.21460
ϕ	4.58471	1.70365	6.17030	1.68461	6.17856	6.20620
t_0	6804.377	6804.506	6804.960	6804.405	6804.950	6805.033
t_E	5.875	5.387	18.626	4.551	18.162	16.409
χ^2	137.174	139.442	139.538	140.092	142.907	143.867

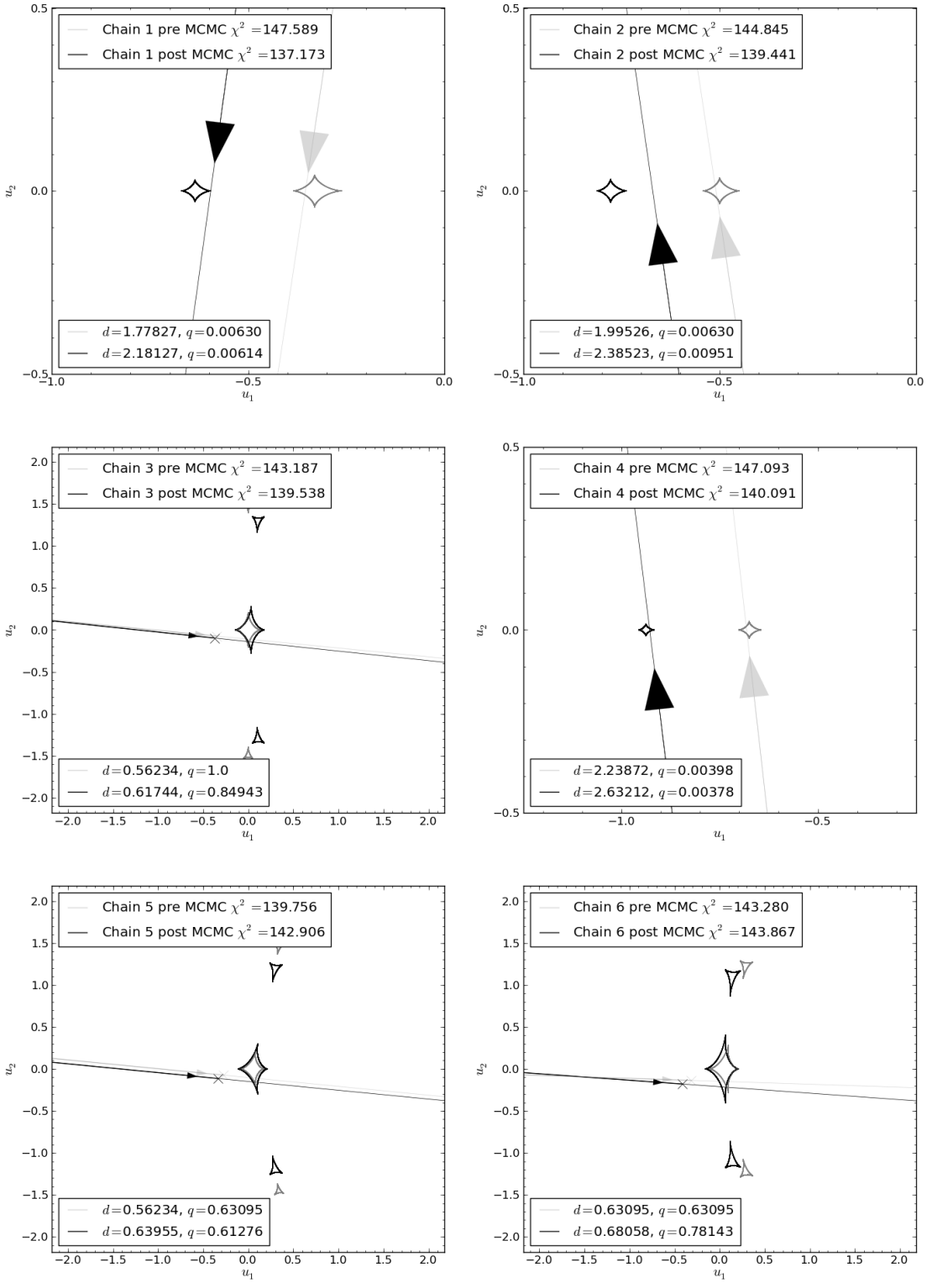


FIGURE 8.27: Local minima MCMC search of a full data set, chain evolution of caustic structures and trajectories (Table 8.12).

Of the six local area MCMC searches the chains that achieved the lowest χ^2 was identified to be in the global minimum area. Taking the last point in this chain, an EMCEE search was performed to fine tune the parameter values and determine the error estimators of the solution. This produced the parameter set seen in Table 8.13, and the caustic with overlaid trajectory, light curve, and residuals shown in Figures 8.28-8.30.

TABLE 8.13: Seven parameter binary lens model solution from a search of a full data set of OGLE-2014-BLG-0945, determined using the GPU modelling method.

Parameter	GPU solution
d	2.23 ± 0.32
q	0.0060 ± 0.0016
ρ	0.128 ± 0.036
u_0	-1.72 ± 0.31
ϕ	4.59 ± 0.93
t_0	6804.47 ± 0.30
t_E	5.63 ± 0.77

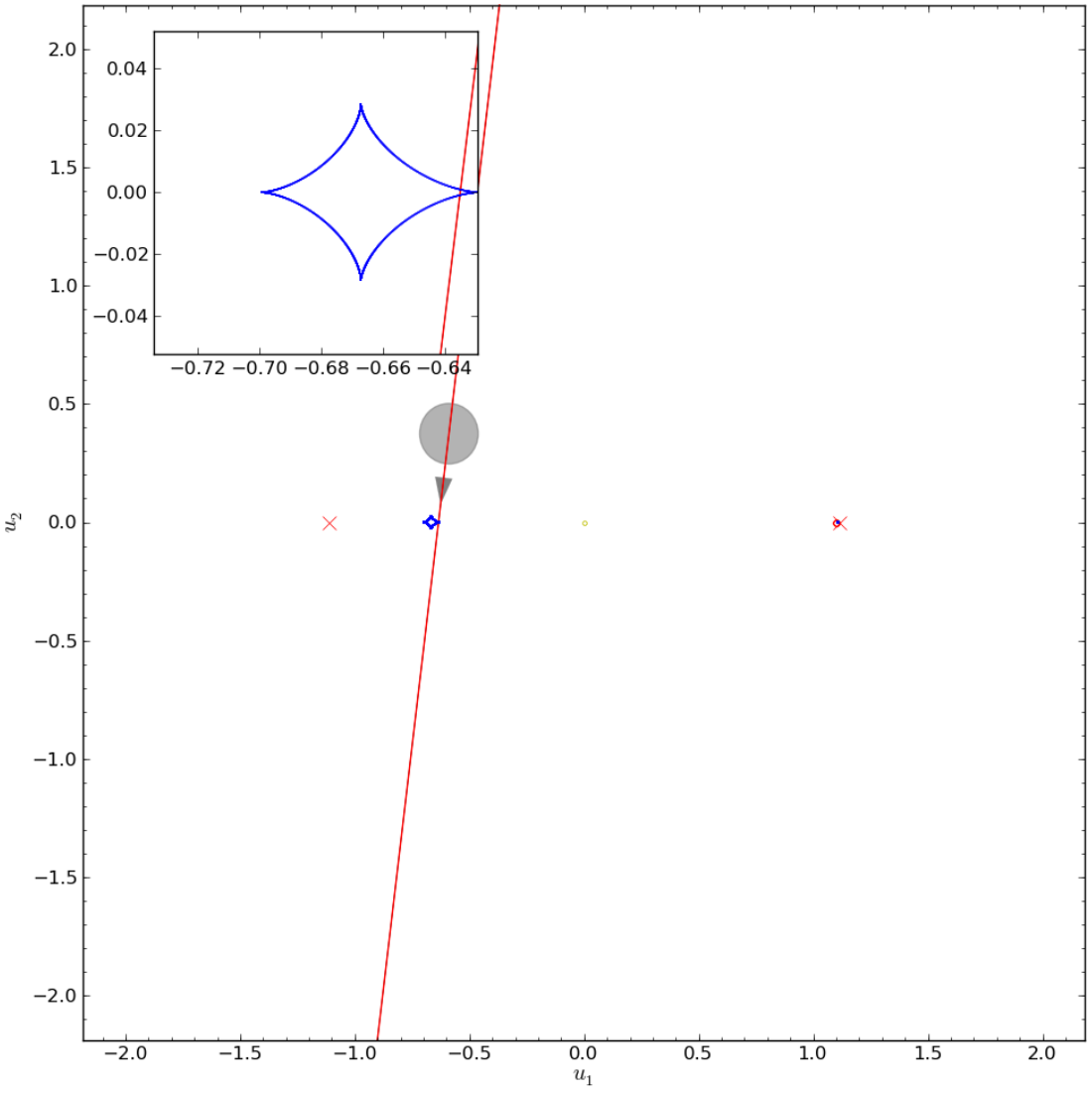


FIGURE 8.28: A caustic map with a overlaid trajectory for the model solution (Table 8.13) determined by the EMCEE GPU search of a full data set of OGLE-2014-BLG-0945.

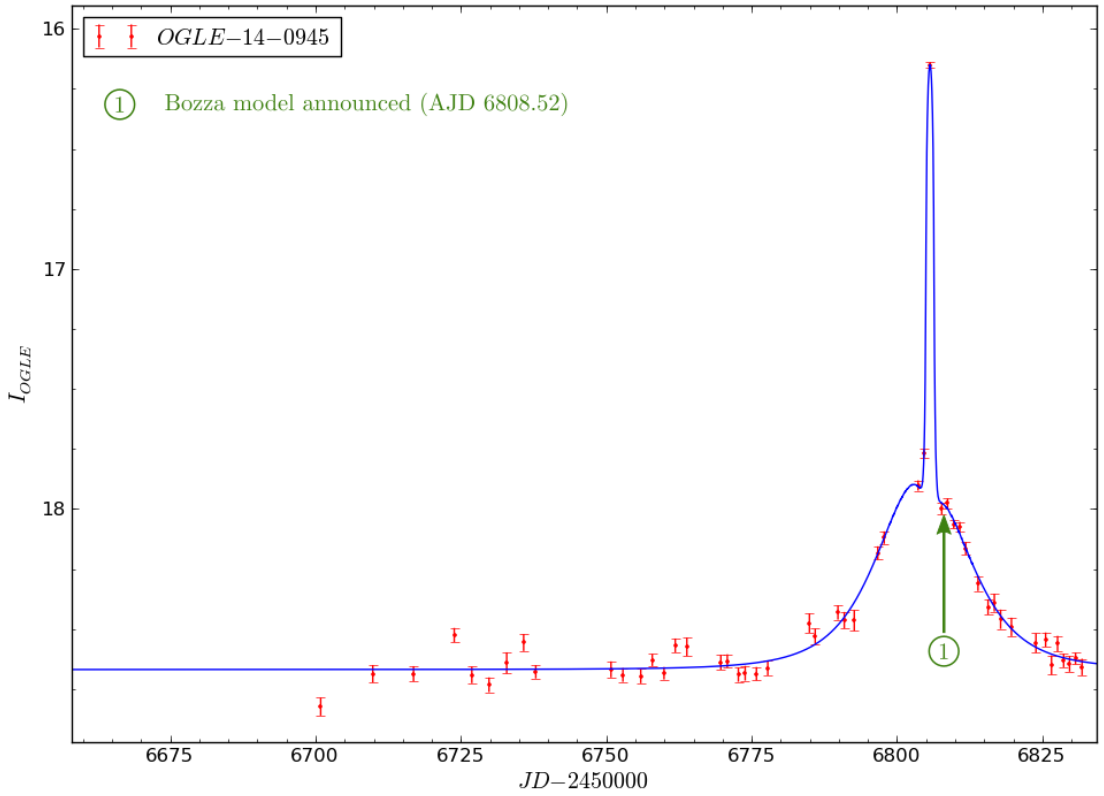


FIGURE 8.29: A light curve of the model solution (Table 8.13) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-0945.

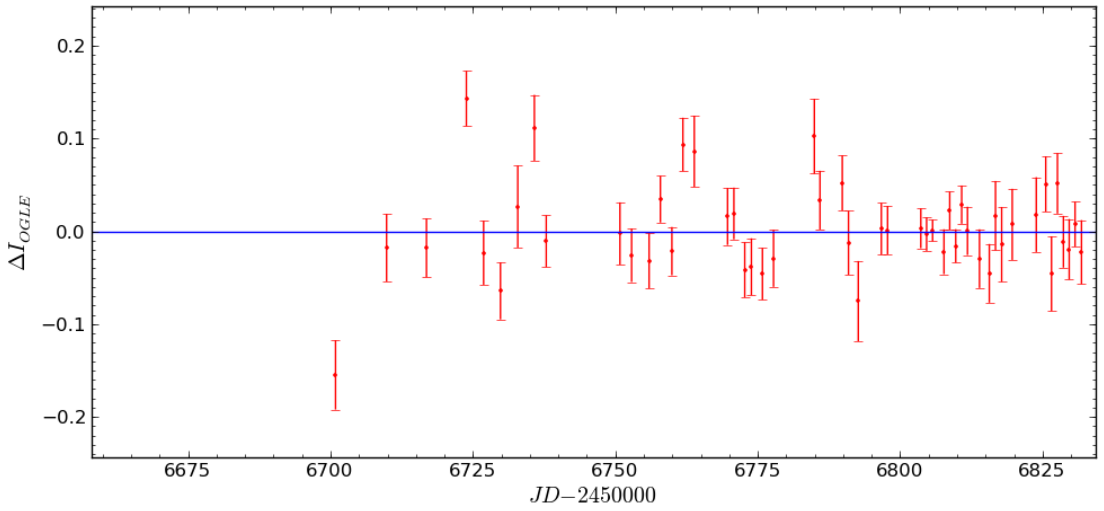


FIGURE 8.30: Light curve residuals of the model solution (Table 8.13) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-0945.

The solution presented here shows agreement with the original solution proposed by V. Bozza, however, with much smaller error terms which may come from the increased quantity of data being able to better constrain the model. Table 8.12 shows that the best chain solution is only favored over a close binary lens solution by $\Delta\chi^2 \approx 2.4$ showing that with the limited data sampling available, no single solution is significantly better than the others presented, thus it is not possible to confirm the planetary candidate solution.

Our analysis is important in this event, as it is able to support other people's proposed models, and in this instance highlight the possible alternate solutions. As a means of checking our solutions, the modelling procedure was re-run three weeks later (AJD 6853.41), including an extended baseline (AJD 6400.0-6853.5). This search resulted in the same global minimum area, but shifted the mass ratio to a slightly larger value of $q \approx 0.013$. With the extra baseline data, in particular the time after the event has passed, it results in a greater constraint in the curvature of the light curve returning to baseline. However, despite the increase of baseline data it is still not possible to suggest any one solution is preferential to the others proposed.

8.6 OGLE-2014-BLG-1102

The event OGLE-2014-BLG-1102 is of very low magnification and was identified to be anomalous by the number of data points that were higher than would be expected for a single lens model. From the raw data it was possible to see that a caustic crossing event had occurred, and the data resided in the trough between the two crossing events. The binary lens modeller was initialized on 2014 June 24, with a requirement for the model to contain a minimum of two peaks located around the AJDs of 6823.5 and 6825.0, and needing to achieve a minimum change in magnitude of 0.05 or more. Our model minimization procedure was performed on 360 OGLE data points between the AJD range of 6750.0 and 6833.4, which completed the grid and six local area MCMC searches in 2 hours and 29 minutes, producing the results in Table 8.14, and Figures 8.31 and 8.32.

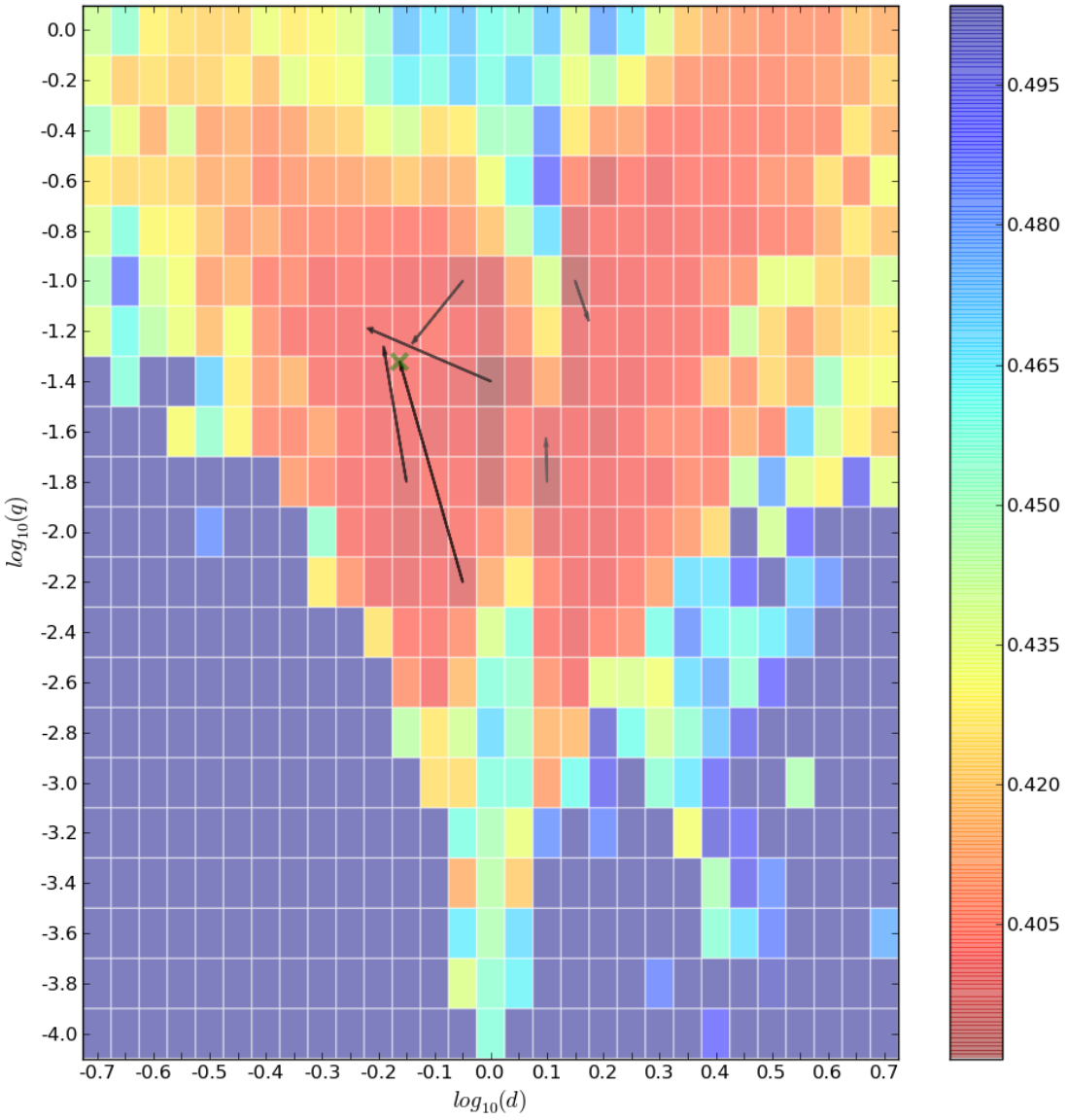


FIGURE 8.31: Local minima MCMC search of a full data set search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2014-BLG-1102.

TABLE 8.14: Seven parameter binary lens local minima model solutions from a search of a full data set of OGLE-2014-BLG-1102, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	0.68948	0.64492	0.60560	0.72645	1.49117	1.25326
q	0.04739	0.05431	0.06462	0.05670	0.06992	0.02343
ρ	6.13×10^{-6}	3.54×10^{-4}	8.32×10^{-4}	7.72×10^{-7}	4.69×10^{-6}	0.00275
u_0	−0.05877	−0.04954	−0.03470	−0.08549	−0.07182	−0.49767
ϕ	4.73793	4.73417	4.74945	4.72053	4.80449	4.70865
t_0	6823.909	6823.942	6823.378	6823.799	6824.158	6824.078
t_E	81.686	83.755	96.664	65.231	49.640	16.119
χ^2	142.505	143.284	143.357	143.388	146.067	148.751

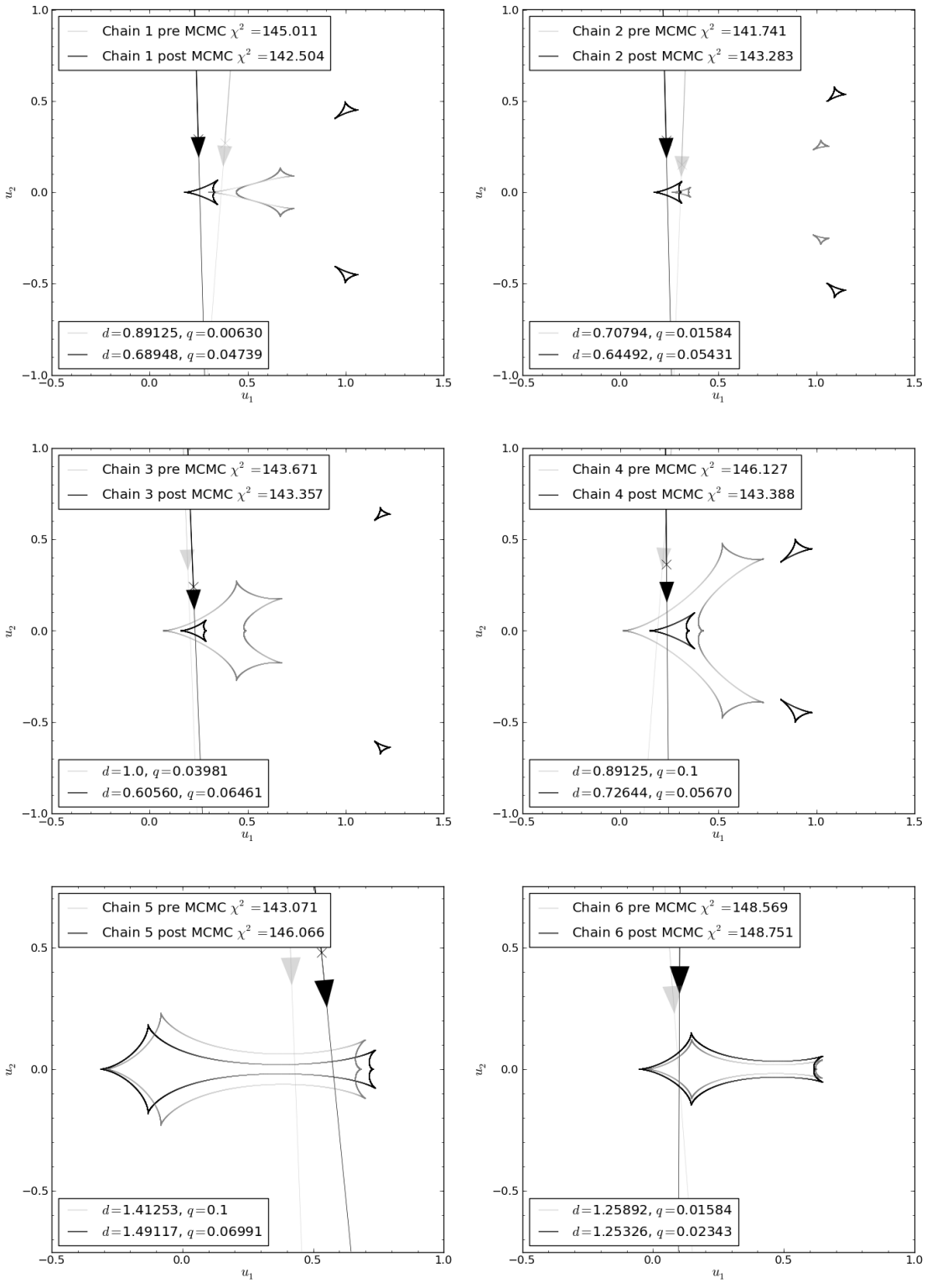


FIGURE 8.32: Local minima MCMC search of a full data set, chain evolution of caustic structures and trajectories (Table 8.14).

The lowest χ^2 chain was used as a starting point for an EMCEE search to fine tune the solution and determine error estimates for each parameter, producing the results shown in Table 8.15 and Figures 8.33-8.35.

TABLE 8.15: Seven parameter binary lens model solution from a search of a full data set of OGLE-2014-BLG-1102, determined using the GPU modelling method.

Parameter	GPU solution
d	0.684 ± 0.056
q	0.049 ± 0.019
ρ	$6.7 \pm 3.9 \times 10^{-6}$
u_0	-0.058 ± 0.017
ϕ	4.68 ± 0.10
t_0	6823.84 ± 0.50
t_E	85 ± 22

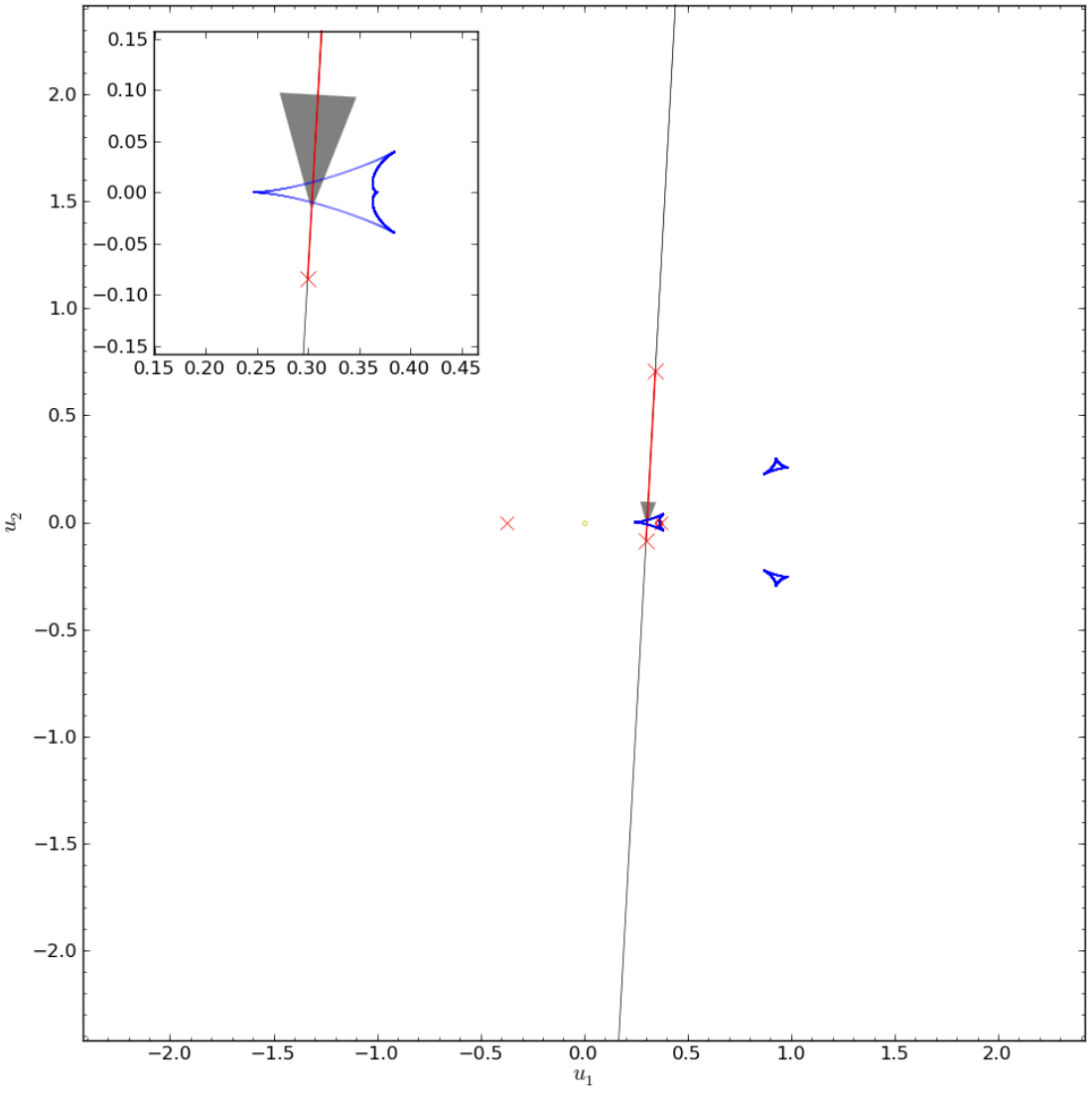


FIGURE 8.33: A caustic map with a overlaid trajectory for the model solution (Table 8.15) determined by the EMCEE GPU search of a full data set of OGLE-2014-BLG-1102.

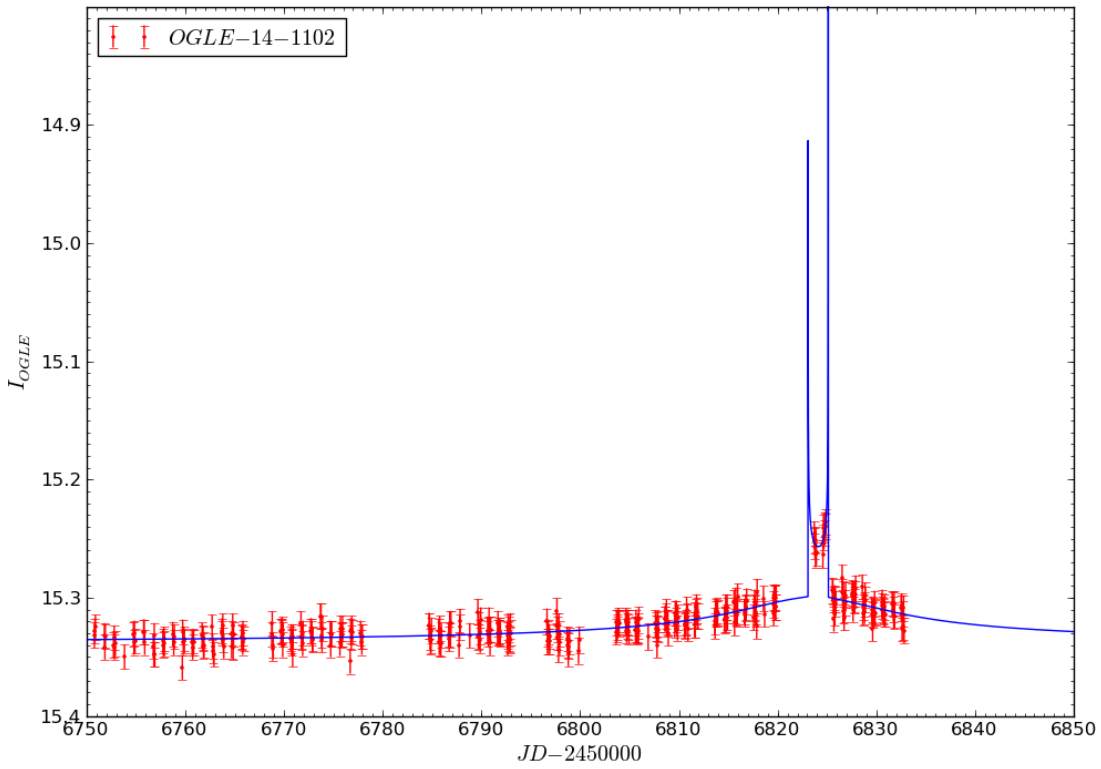


FIGURE 8.34: A light curve of the model solution (Table 8.15) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-1102.

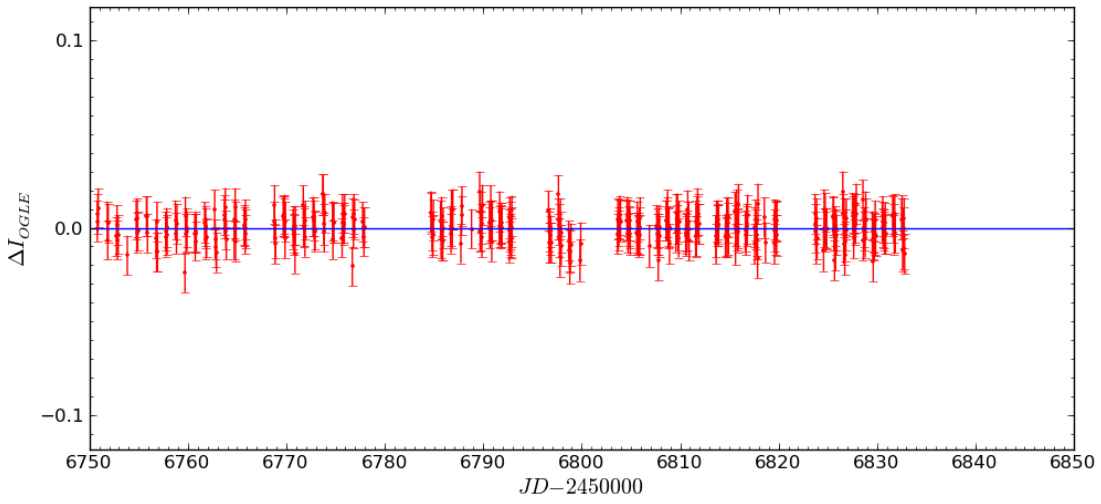


FIGURE 8.35: Light curve residuals of the model solution (Table 8.15) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-1102.

Three days later an announcement to the microlensing community was made by V. Bozza (V. Bozza, e-mail to ulens-analysis mail list on 2014 June 27 AJD 6836.87) who proposed a pair of solutions, one static wide binary star and a closer planetary solution that includes parallax, concluding that the planetary solution was more likely. In response to this, our model was circulated showing our brown dwarf solution. This brown dwarf solution shows agreement with the close solution of V. Bozza, yet with the event being poorly sampled it is not possible to confirm that a planetary solution is significantly better than any of the alternate binary lens solutions presented in Table 8.14.

Three weeks later (2014 July 15, AJD 6853.40) a re-modelling of this event was initialized with the latest data, including the events return to baseline. The d, q grid search produced a minimized χ^2 map and located similar regions of parameter space to explore with MCMC as before. This produced five unique solutions (from the six chains) including the planetary solution previously identified, all within a range of $\Delta\chi^2 < 5$. The re-analysis shows that despite the inclusion of data that covers the full length of the event, the poorly sampled caustic crossing feature prevents any model being preferential to the others.

8.7 OGLE-2014-BLG-1327

The faint event OGLE-2014-BLG-1327 (baseline magnitude of 20.17) was following the typical shape of a predicted single lens model, until on the descent after the peak a few points started to rise again unexpectedly. With noisy data and a small number of observations showing the anomaly, it hard to visually estimate the cause of the perturbation. Therefore our modelling procedure was initialized on the 2014 July 13 (AJD 6851.42) to identify the cause. Our modelling procedure was provided with all available data between the AJDs of 6600 and 6850, which totalled 62 OGLE data points. The model was required to achieve a minimum change in magnitude of at least 1.5, and contain two peaks or more with one being located close to 6841.35. An initial grid search took 58 minutes and 57 seconds to complete, from which six local MCMC searches were started to further explore the parameter space and check for degeneracies. A total of 2 hours and 3 minutes were taken to complete the full search, producing the results shown in Table 8.16, and Figures 8.36 and 8.37.

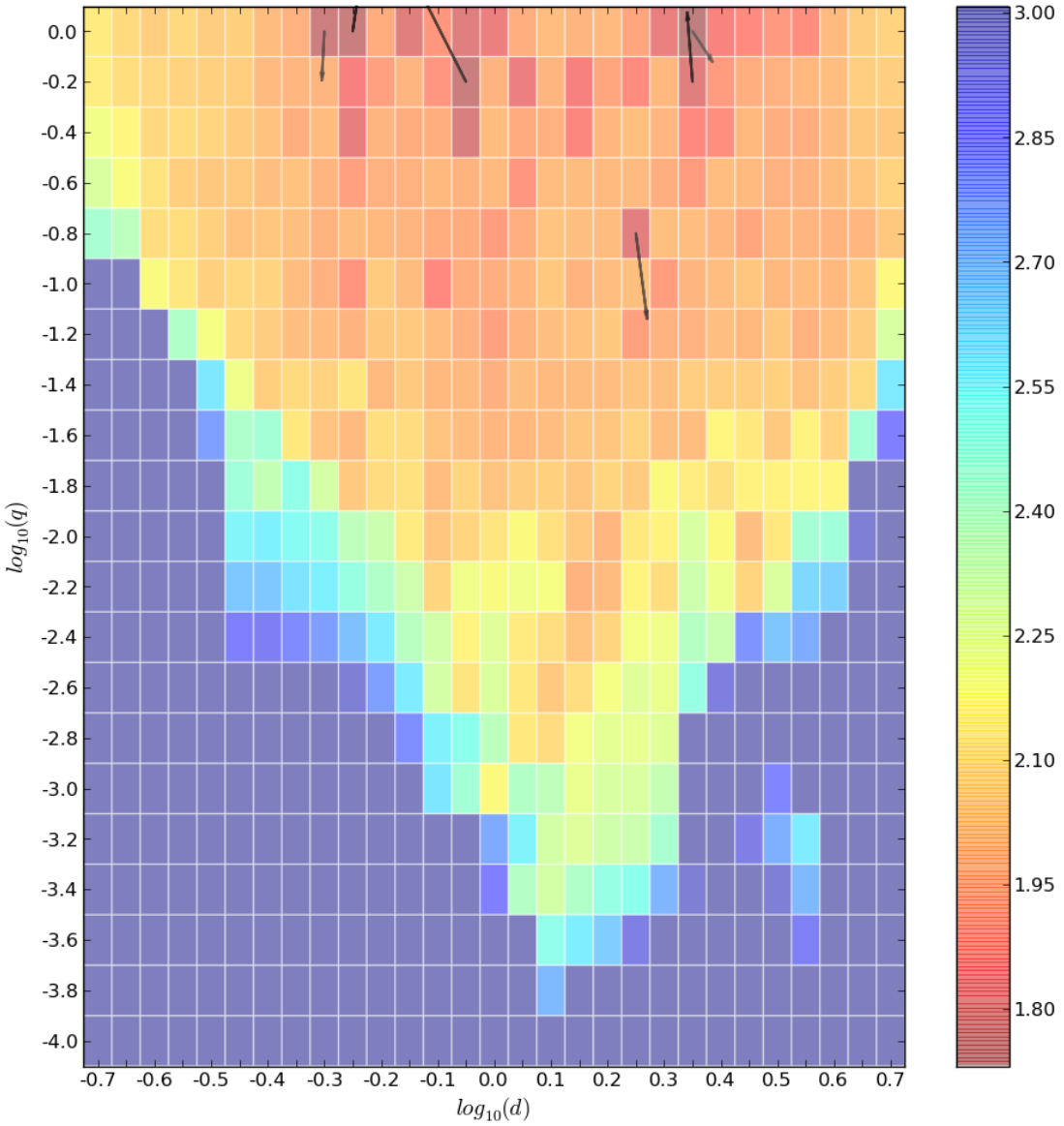


FIGURE 8.36: Local minima MCMC search of a full data set search movement overlaid onto the initial minimized χ^2 d, q map of OGLE-2014-BLG-1327.

TABLE 8.16: Seven parameter binary lens local minima model solutions from a search of a full data set of OGLE-2014-BLG-1327, determined using the GPU modelling method.

Parameter	Chain 1	Chain 2	Chain 3	Chain 4	Chain 5	Chain 6
d	0.59389	2.19066	0.65937	1.86121	0.49611	2.42709
q	2.24538	1.18670	2.40530	0.07284	0.64194	0.75920
ρ	0.04198	0.04149	0.02932	0.01371	9.455×10^{-7}	0.02800
u_0	−0.18634	−0.78169	−0.11462	−0.00364	−0.08243	−0.76456
ϕ	4.24196	4.07296	2.64872	3.59107	4.22790	2.20584
t_0	6846.669	6828.706	6848.269	6848.129	6846.166	6874.558
t_E	23.272	35.417	36.563	60.886	27.464	59.934
χ^2	111.157	112.314	113.186	113.974	115.092	118.661

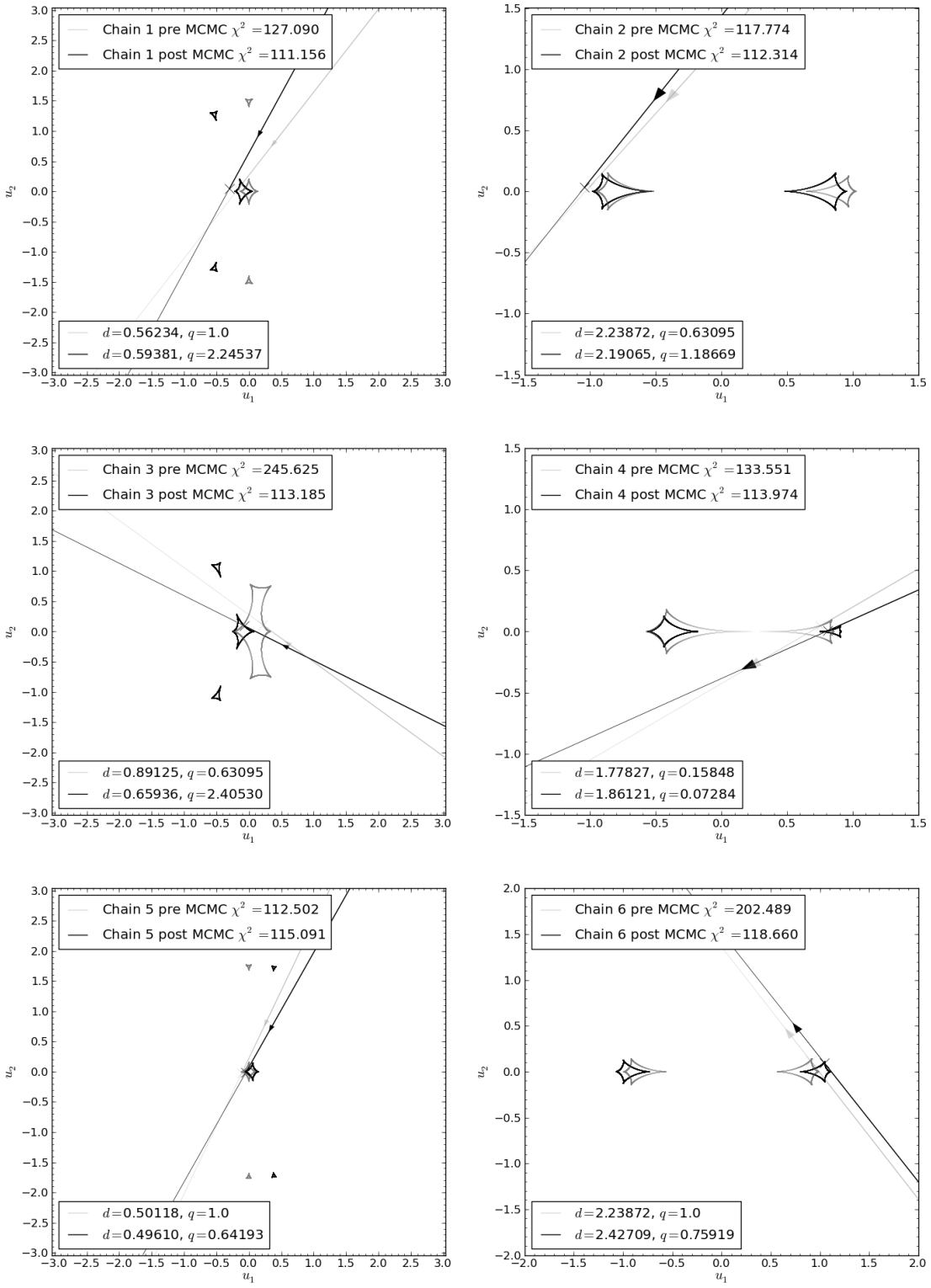


FIGURE 8.37: Local minima MCMC search of a full data set, chain evolution of caustic structures and trajectories (Table 8.16).

Of the six locally explored MCMC searches the chain that produced the lowest χ^2 was determined to have located the area of global minimum. This lead to the initialization of an EMCEE search to fine tune the solution set and determine an error term for each parameter. The burn in and post burn in sample search took a total of 1 hour 34 minutes, and produced the parameter set shown in Table 8.17, which produces the caustics and trajectory, light curve, and residuals shown in Figures 8.38-8.40.

TABLE 8.17: Seven parameter binary lens model solution from a search of a full data set of OGLE-2014-BLG-1327, determined using the GPU modelling method.

Parameter	GPU solution
d	0.615 ± 0.065
q	3.3 ± 1.1
ρ	0.047 ± 0.015
u_0	-0.149 ± 0.037
ϕ	4.26 ± 0.22
t_0	6846.3 ± 1.4
t_E	29.0 ± 4.1

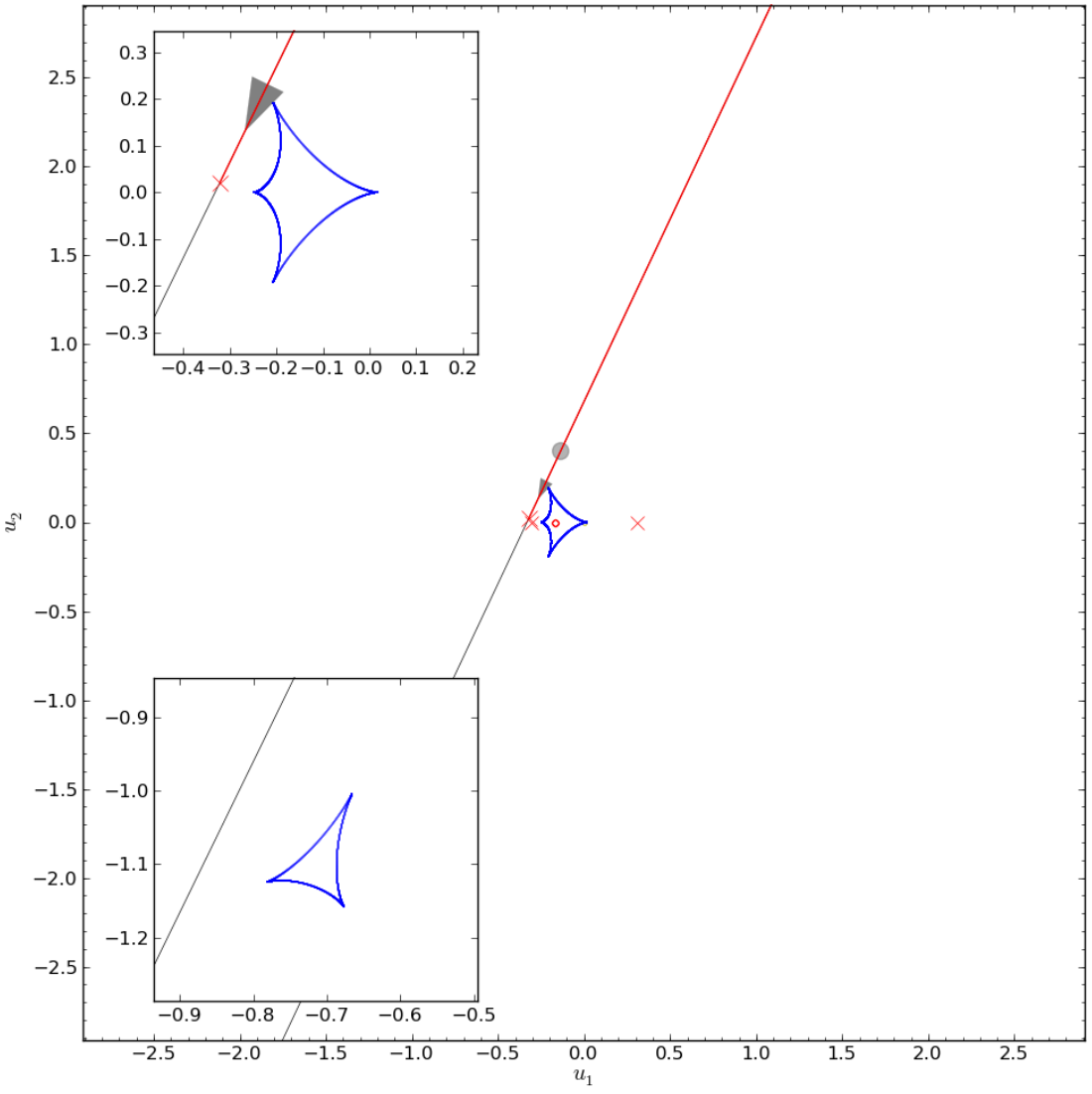


FIGURE 8.38: A caustic map with a overlaid trajectory for the model solution (Table 8.17) determined by the EMCEE GPU search of a full data set of OGLE-2014-BLG-1327.

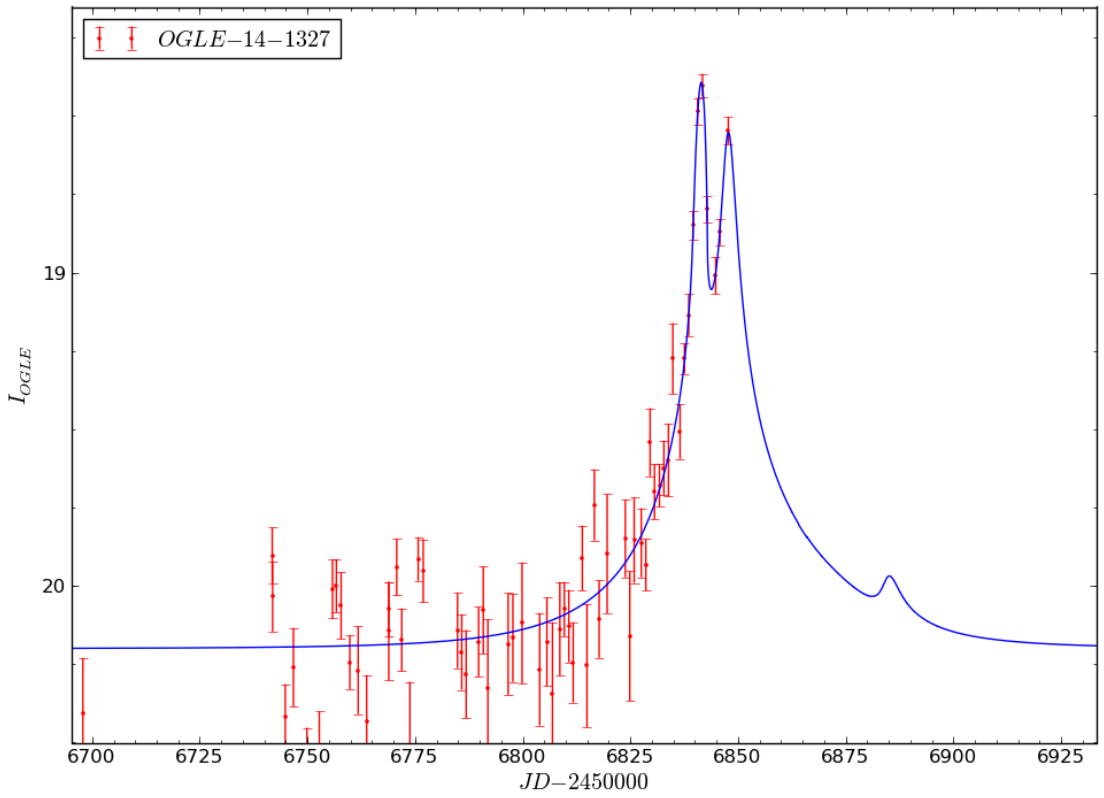


FIGURE 8.39: A light curve of the model solution (Table 8.17) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-1327.

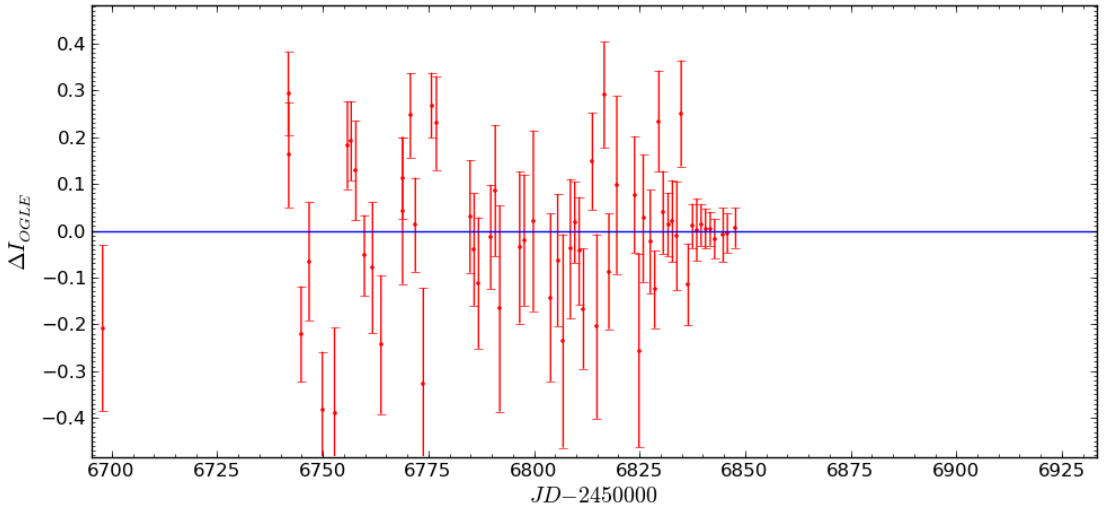


FIGURE 8.40: Light curve residuals of the model solution (Table 8.17) determined by the GPU EMCEE search method on a full data set of OGLE-2014-BLG-1327.

The solution shown in 8.17 indicates that a binary lens system is currently the most likely cause for the anomaly detected in this event. However, it can be seen that the constraint on the accuracy of the mass ratio parameter is not optimal, which is corrolated to the large error terms in t_0 and t_E. This lack of constraint can also be seen by the broad areas of minima in the d, q grid search, and the large chain movement in the local MCMC search and EMCEE exploration. The poor constraint on the q parameter may come from the small amount of data that describes the anomaly and the large error in this data due to the event being faint, including that the event is still ongoing and the second peak is unconstrained by any data (as it is still rising).

Although the binary lens solution is not solved to high precision, it is clear from the grid search that no potential planetary models produce viable solutions. This resulted in the announcement of our model to the microlensing community on 2014 July 13, stating the binary lens model found.

Modelling all anomalous events as soon as possible and as quickly as possible, such as this one, is important so that observers know what could have caused the anomaly, and whether follow up observations of the event are desired. The grid search and local area MCMC explorations of this event completed in about 2 hours, providing a quick response from initializing the model to interpreting it. A quick analysis allows for an e-mail alert to the microlensing community to be announced without too much delay, helping prompt decisions to be made about observational target choices.

8.8 Summary

Now that the GPU model minimization method presented in this thesis has been validated in the previous chapter, it is being used during the current season to model ongoing binary lens events. To date it has modelled 68 unique events[1], with the latest model solutions displayed in Table 8.18 listing 23 planetary solutions, 23 brown dwarf solutions, and 22 stellar solutions. For each event, the table identifies the solution's binary companion as Stellar, Brown Dwarf, or Planetary based on the mass ratio of the best solution determined. Assuming a typical lens mass to be $0.25M_{\odot}$, and a brown dwarf to have a mass range of $0.013M_{\odot}$ to $0.072M_{\odot}$, the mass ratio must be $0.052 < q < 0.28$ to equal a brown dwarf. Greater than this range implies a stellar companion, whereas less than implies planetary. All solutions in Table 8.18 show the latest best fitting solution, but do not necessarily take into consideration the amount or state of the data used, or possible degenerate solutions. A column of notes is provided with each model to identify

[1] As of 2014 August 13

important details about the data of each event, and what future work may be required to confirm the model solutions.

Six of the events from Table 8.18 are detailed in this chapter, where it presents MOA-2014-BLG-0075/OGLE-2014-BLG-0241 and OGLE-2014-BLG-0945 as star-planetary events, and OGLE-2014-BLG-1102 as a possible star-brown dwarf lens. The remaining three events (MOA-2014-BLG-153/OGLE-2014-BLG-0621, MOA-2014-BLG-0205/OGLE-2014-BLG-0626, and OGLE-2014-BLG-1327) are all identified to be star-star binaries.

This chapter shows the importance of using a rapid real-time modelling system as early as possible for current events. Our system enabled alerts to be sent out to the microlensing community that proposed a potentially new planetary candidate event, and advised against an event being a planetary candidate solution, which had previously been proposed by others. An outline of the modelling performed for each event as new data was acquired is shown in chronological order. The success of all of these modelled events is apparent from the agreement of other modelling groups confirming the solutions we presented via private communications to the microlensing community. It can also be seen that the analysis of just the lowest χ^2 solution is insufficient in determining the model, and the need to explore several minima is essential to ensure that the global solutions is identified and confirmed to be significantly better than any other local minimum.

TABLE 8.18: All binary lens models completed with the method outlined in this thesis for the 2014 season, where each event is identified in the first column (the year, 2014, and target area, BLG, have been dropped to save space). The second column indicates if the solution is a star-star, star-brown dwarf, or star-planetary solution. Following the model solution is a column of notes (detailed in Table 8.19) identifying the state of the data and any possible further work which may be required.

Event ID	Model Solution	Comments / Further action required
MOA-0029	Stellar	
MOA-0032/OGLE-0062	Brown Dwarf	Re-reduce
MOA-0044	Stellar	Noisy, Poorly sampled
MOA-0047 / OGLE-0104	Stellar	SL FS compare
MOA-0048 / OGLE-0135	Brown Dwarf	Re-reduce
MOA-0051 / OGLE-0251	Planetary	Re-reduce
MOA-0056 / OGLE-0159	Stellar	
MOA-0065	Planetary	SL FS compare
MOA-0066 / OGLE-0042	Stellar	
MOA-0075 / OGLE-0241	Planetary	Re-reduce
MOA-0117 / OGLE-0270	Brown Dwarf	Update dev
MOA-0147 / OGLE-0383	Planetary	SL FS compare
MOA-0148 / OGLE-0257	Brown Dwarf	Update agr
MOA-0153 / OGLE-0621	Stellar	
MOA-0159 / OGLE-0550	Brown Dwarf	Update agr
MOA-0165 / OGLE-0572	Brown Dwarf	Update agr
MOA-0171 / OGLE-0319	Planetary	Re-reduce
MOA-0198 / OGLE-0764	Planetary	SL FS compare
MOA-0205 / OGLE-0626	Stellar	
MOA-0222 / OGLE-0706	Planetary	Noisy baseline, Poorly constrained
MOA-0230 / OGLE-0536	Brown Dwarf	Re-reduce, Update agr
MOA-0245 / OGLE-0787	Planetary	SL FS compare
MOA-0260	Stellar	Update dev
MOA-0270	Stellar	Update agr
MOA-0285 / OGLE-0962	Brown Dwarf	Update dev
MOA-0290 / OGLE-0914	Planetary	Re-reduce, Update agr
MOA-0305 / OGLE-0900	Stellar	Update agr
MOA-0307 / OGLE-0124	Planetary	Update agr
MOA-0347	Planetary	Update agr
MOA-0367 / OGLE-1255	Brown Dwarf	Re-reduce
MOA-0368 / OGLE-1112	Planetary	SL FS compare, Update agr
MOA-0369	Planetary	SL FS compare, Update agr
MOA-0373 / OGLE-1407	Planetary	SL FS compare, Update agr
MOA-0399 / OGLE-1156	Planetary	Update dev
MOA-0405 / OGLE-1479	Stellar	Ongoing
MOA-0438 / OGLE-1507	Planetary	Ongoing
OGLE-0074	Stellar	SL FS compare, Update agr
OGLE-0144	Planetary	Update agr
OGLE-0185	Brown Dwarf	Ongoing
OGLE-0188	Brown Dwarf	Re-reduce
OGLE-0213	Planetary	Re-reduce, Update agr
OGLE-0215	Brown Dwarf	Poorly sampled, Ongoing
OGLE-0259	Stellar	Update agr

Event ID	Model Solution	Comments / Further action required
OGLE-0290	Stellar	Poorly sampled, Update agr
OGLE-0298	Brown Dwarf	Ongoing
OGLE-0355	Brown Dwarf	Re-reduce
OGLE-0431	Stellar	Update agr
OGLE-0475	Stellar	Update dev
OGLE-0497	Stellar	Re-reduce, Poorly sampled
OGLE-0539	Brown Dwarf	Update dev
OGLE-0589	Planetary	SL FS compare, Update agr
OGLE-0650	Stellar	Update agr
OGLE-0657	Brown Dwarf	Noisy, Re-reduce, Update agr
OGLE-0790	Brown Dwarf	Update
OGLE-0801	Planetary	SL FS compare
OGLE-0871	Stellar	Very Noisy, Re-reduce
OGLE-0935	Stellar	SL FS compare
OGLE-0945	Planetary	
OGLE-0989	Brown Dwarf	Poorly sampled
OGLE-1050	Brown Dwarf	Update agr
OGLE-1102	Brown Dwarf	Poorly constrained, Re-reduce, Planetary and stellar solutions are viable
OGLE-1126	Brown Dwarf	Re-reduce, Update agr
OGLE-1136	Stellar	Ongoing
OGLE-1168	Brown Dwarf	Re-reduce, Update agr
OGLE-1239	Brown Dwarf	Comparable planetary solution, Re-reduce, Cosmic ray
OGLE-1327	Stellar	Update agr, Poorly sampled, Re-reduce
OGLE-1413	Planetary	SL FS compare, Update agr
OGLE-1437	Planetary	Systematic errors in data, false planet

TABLE 8.19: Descriptions of the notation used in Table 8.18.

Note	Definition
Cosmic ray	Data is known to contain a cosmic ray, which needs removing from the data set.
Noisy	The source data is very noisy.
Ongoing	The data is in agreement with the model, but the event is still ongoing.
Poorly sampled	The observational cadence is too low to constrain the model well.
Re-reduce	The data should be re-reduced before being re-run through the modeller to check the solution.
SL SF compare	The model should be compared against a single lens finite source model to validate the binary lens result.
Update agr	New data exists that was not included in the modelling of this event, and it follows the predicted light curve of the current best solution.
Update dev	New data exists that was not included in the modelling of this event, and it does not follow the current best solution.

Chapter 9

Summary and outlook

9.1 Summary

Starting from the historical origins and mathematical derivations that determine the magnification of a microlensing event, an overview of currently used methods and challenges faced when modelling this phenomenon has been shown. A new modelling system is presented in the preceding pages, which performs a single lens microlensing analysis of all recently discovered events. In addition, a binary lens modelling process has been produced to search a wide parameter space and identify a global solution, given minimal input from a user. The detailed method to modelling binary lens events utilizes the large-scale parallel abilities of NVIDIA CUDA GPUs, and new search strategies that have been developed to locate the global parameter solution in a short time. Optimizations have been applied when ever possible, achieving performance gains up to 1200 times, compared with standard CPU methods.

The need of a rapid modelling procedure has already been highlighted in Chapter 1, which has been addressed through the work presented here. The results of Chapter 7 show that, within short time frames, our method is able to confirm existing models presented in published literature and determine new viable solutions. Most notably of these is the analysis of MOA-2003-BLG-0053/OGLE-2003-BLG-0235 in Section 7.3 where a thoroughly studied event is presented, including the time taken to identify the global solution by a different modelling technique. It is shown that not only does our method achieve the same global solution area, but it does so in a shorter time (25% faster), despite searching a larger solution space (over 117 times greater). Although no other events presented include time comparisons, all solutions in Section 7.2 (with one exception) are shown to complete a parameter search in a short time, with an average of a little over two hours.

Optimizing the modelling processes has been a large component of our presented work, which achieves the rapid modelling times. Although optimizing the modelling process is the preferential way to improving performance, due to not requiring additional resources, the option of improved hardware still exists. As stated earlier, alternate methods of making and storing the magnification maps may reduce computation time, but current hardware availability prevents this being a viable method. Similarly the continual development of new GPU hardware can lead to faster processing, and more parallel possibilities, which would perform the same analysis faster. This has already been investigated and found to provide a four times performance improvement (on an NVIDIA Tesla K40) in some of the analysis, but was unable to run the same code for other parts. To utilize alternate hardware solutions, additional work will be required to ensure the code performs as expected on different CUDA devices. Not only can the performance be improved with the use of newer GPUs, but also the possibilities of multiple GPUs working in parallel. GPU clusters exist, that if utilized properly could allow every pixel of the grid search to be performed in parallel. Therefore making the entire grid search process take around ten seconds. Multiple GPUs would also allow a greater number of local area MCMC searches to be performed in parallel. This would not only shorten the analysis time, but also enable a denser analysis of parameter space without any time penalties, helping ensure no solutions are missed.

Our proven method is currently being used to study current events in the 2014 microlensing season (Figure 9.1), during which it has identified a new planetary candidate (Section 8.2), and disproved another (Section 8.4), among the many more events which are continually being modeled and published through the web site developed (Section 6). A rapid modelling system enables a thorough analysis of a large number of microlensing events with minimal input from a user. This allows us to study the latest events, and quickly send out alerts to the microlensing community when interesting discoveries are made, as was the case with MOA-2014-BLG-0075/OGLE-2014-BLG-0241 and MOA-2014-BLG-0205/OGLE-2014-BLG-0626. By creating a web site that shows the live modelling of microlensing events, it enables users to check the previous analysis and monitor the latest modelling progress to see if a planetary candidate may have caused an on-going event. Such a system is vital in helping allocate the limited observing resources that are available. With the growing number of microlensing detections each year, it is essential that an accurate model can be produced as quickly as possible to help prioritize which targets should be followed up with intensive observations.

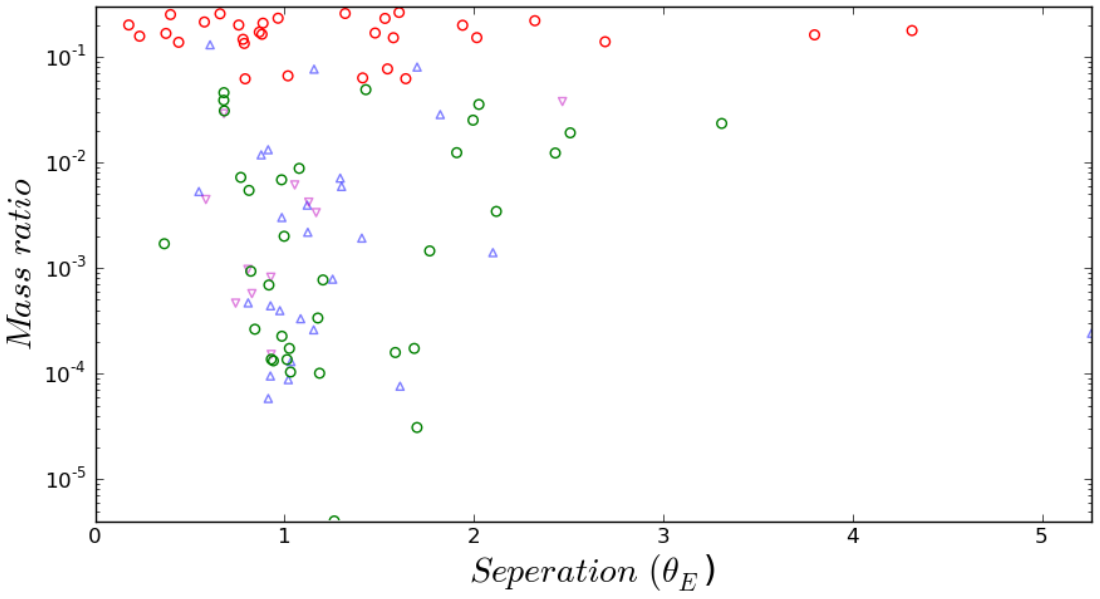


FIGURE 9.1: The distributions of the system's proposed planetary (green circles) and brown dwarf (red circles) detections on the web site on 2014 November 17. This distribution is based on the database's best fitting solution and does not consider the quantity or quality of the input data, or any degenerate solutions. For comparison, the known exoplanets found via microlensing (from The Extrasolar Planets Encyclopaedia) are included in the plot as blue triangles, and a list of announced microlensing detections in 2014 by C. Han (C. Han, e-mail to the ulens-analysis mail list on 2014 October 25) are shown by inverted pink triangles (some of these will be duplicates of our system's detections).

9.2 Limitations

The method detailed here is shown to successfully locate the global solutions of all but one of the events presented. However, challenges are known to exist when dealing with parameter sets that produce very small caustic structures. The finite resolution of the magnification maps results in a lower limit to the source size. This can only be resolved by improving the resolution of the magnification maps, a process that comes with a heavy time penalty. Improved hardware will likely be required to resolve this issue before improved map accuracy can feasibly be implemented. As with any grid search method a limitation in a model's accuracy comes from the resolution of the parameter grid. With the currently implemented grid search method, a single u_0, ϕ trajectory is taken as the local minimum of each d, q pixel. This may result in potential solutions being overlooked as an alternate local minimum may ultimately lead to a better solution once d, q are not constrained by the grid resolution. Such limitations have been explored, and via experimentation the currently used method is chosen due to its rapid results with minimal compromises.

9.3 Future work

The main result presented here has been the development of a rapid modelling system for binary lens events. This lead to the production of a web site and control interface to help users study events without requiring any knowledge of the methods used or the computer network system that the modelling processes operate on. The web site interface could be further improved by providing a user with a greater quantity of information about the past and current modelling of any particular event. These improvements would enable users to visualize all of the minimized local area MCMC searches so comparisons with the global minimum can be made, and the predicted light curves of alternate, possibly viable, solutions can be seen.

Further improvements would result in the migration of the modelling system's control software into the web site, allowing users to initialize models and analyze data from any internet accessible device. This would make the control software simpler to use and enables the option to provide visual feedback to the user, which is currently unachievable with the text based control software. Not only can the web site front end be developed further, but the data retrieval system can be enhanced by tools such as R-sync. Additional processes can be put in place to ensure upon modelling that the latest available data is always used.

Improvements in the interface side of the modelling system would help manage, monitor, and lead to an easier user experience that can result in an increased detection rate of interesting events. The addition of more analysis tools are also desirable, such as the inclusion of higher order effects. An option to explore parallax, orbital motion and other effects would provide a greater quantity of information about a solution, potentially leading to learning more about the binary system, such as planetary masses.

Single lens events can be studied further than the current analysis allows, with an inclusion of finite sources and binary sources. The development of these analysis tools would help validate binary lens model solutions, when possible competing single lens models with higher order effects need to be ruled out.

After a single and binary lens modelling system, the next step would be to incorporate a triple lens modelling process. This is a very computationally-intense task and would take a long time to process, despite the rapid methods presented here. New search strategies would be required and further optimized modelling processes would be essential in producing a triple lens model minimization routine.

An addition to the modelling system that is currently in development and being trialed, is an automated anomaly detector. This system would be implemented to check all single

lens events, attempting to identify if any deviation from the model exists and estimate the possible cause. This could lead to automatically attempting to fit finite source single lens models, binary source models, or even make an estimate of the input parameters provided to the binary lens modeler. Careful development of this system is required to prevent an automated system that triggers too many binary lens modeler requests, saturating the limited resources of the modelling system without any prioritization. Even without the ability to automatically initialize binary lens modelling, a list of potential anomalies should be created, to highlight events to a user who can filter out false positives and, when believed appropriate, initialize the modelling process. The anomaly detector does not have to be constrained to looking for anomalies in the single lens model, but a similar system can also be incorporated into the binary lens model. Interpreting perturbations of a binary lens model is more complex than single lens models, as they could be produced from a wider range of causes, such as a poorly solved model, higher order effects that are not included, or even a triple lens system.

With the advancement of computational technologies, new hardware solutions may exist in the near future enabling the modelling process to perform much quicker. Alternate technologies such as solid-state storage should be investigated as should alternate GPU devices. This requires further development of the current CUDA implemented modelling method to ensure it works on other devices. As shown in this thesis, the use of a single GPU produces notably faster results, whereas the use of a GPU cluster could produce enormous performance gains. However, this solution is not financially viable to most, as the cost of a NVIDIA Tesla C2075 GPU cluster large enough that a single GPU only computes a single pixel in the grid search (so that it is able to process the entire initial grid search in ≈ 10 seconds), is over 1.2 million USD[1].

Future work to develop this modelling system will enable users easy access to a wide range of analysis tools and thorough modelling processes, with a minimal learning curve an no requirement to understand the system that it runs on. Such a system is crucial to help study the ever-growing quantity of microlensing events, and provide vital information to observing programs, enabling the prioritization of anomalous events with the goal of identifying more planetary detections.

[1] Calculated with a unit price of ≈$2000 per Nvidia Tesla C2075, and a requirement for 609 devices.

Appendix A

Single lens magnification

The formalism below describes the steps used to derive the magnification of a single lens event. If the observer views a small part of a finite source at an angle β from the lens with an angular width of ϕ subtended from the lens. The corresponding image section would appear in the sky to be an angle θ from the lens and also subtend an angle of ϕ from the lens. From Liouville's theorem that the brightness is conserved during a lensing event, the magnification can be calculated as the ratio of the area of the source and the area of the image, as shown in figure A.1. From the geometry of the areas in Figure A.1, the lengths of the edges are

$$\mathrm{d}u_1 = \beta \mathrm{d}\phi \quad , \quad \mathrm{d}u_2 = \mathrm{d}\beta \tag{A.1}$$

$$\mathrm{d}y_1 = \theta \mathrm{d}\phi \quad , \quad \mathrm{d}y_2 = \mathrm{d}\theta \ . \tag{A.2}$$

Taking the determinant of a Jacobian matrix gives the factor by which a function changes. From the above dimensions the Jacobian matrix is found to be

$$J = \begin{bmatrix} \frac{\beta \mathrm{d}\phi}{\theta \mathrm{d}\phi} & \frac{\beta \mathrm{d}\phi}{\mathrm{d}\theta} \\ \frac{\mathrm{d}\beta}{\theta \mathrm{d}\phi} & \frac{\mathrm{d}\beta}{\mathrm{d}\theta} \end{bmatrix} \ . \tag{A.3}$$

The determinant is taken which reduces down to a single term

$$detJ = \frac{\beta \mathrm{d}\phi}{\theta \mathrm{d}\phi} \frac{\mathrm{d}\beta}{\mathrm{d}\theta} - \frac{\beta \mathrm{d}\phi}{\mathrm{d}\theta} \frac{\mathrm{d}\beta}{\theta \mathrm{d}\phi} \Rightarrow -\frac{\beta}{\theta} \frac{\mathrm{d}\beta}{\mathrm{d}\theta} \ . \tag{A.4}$$

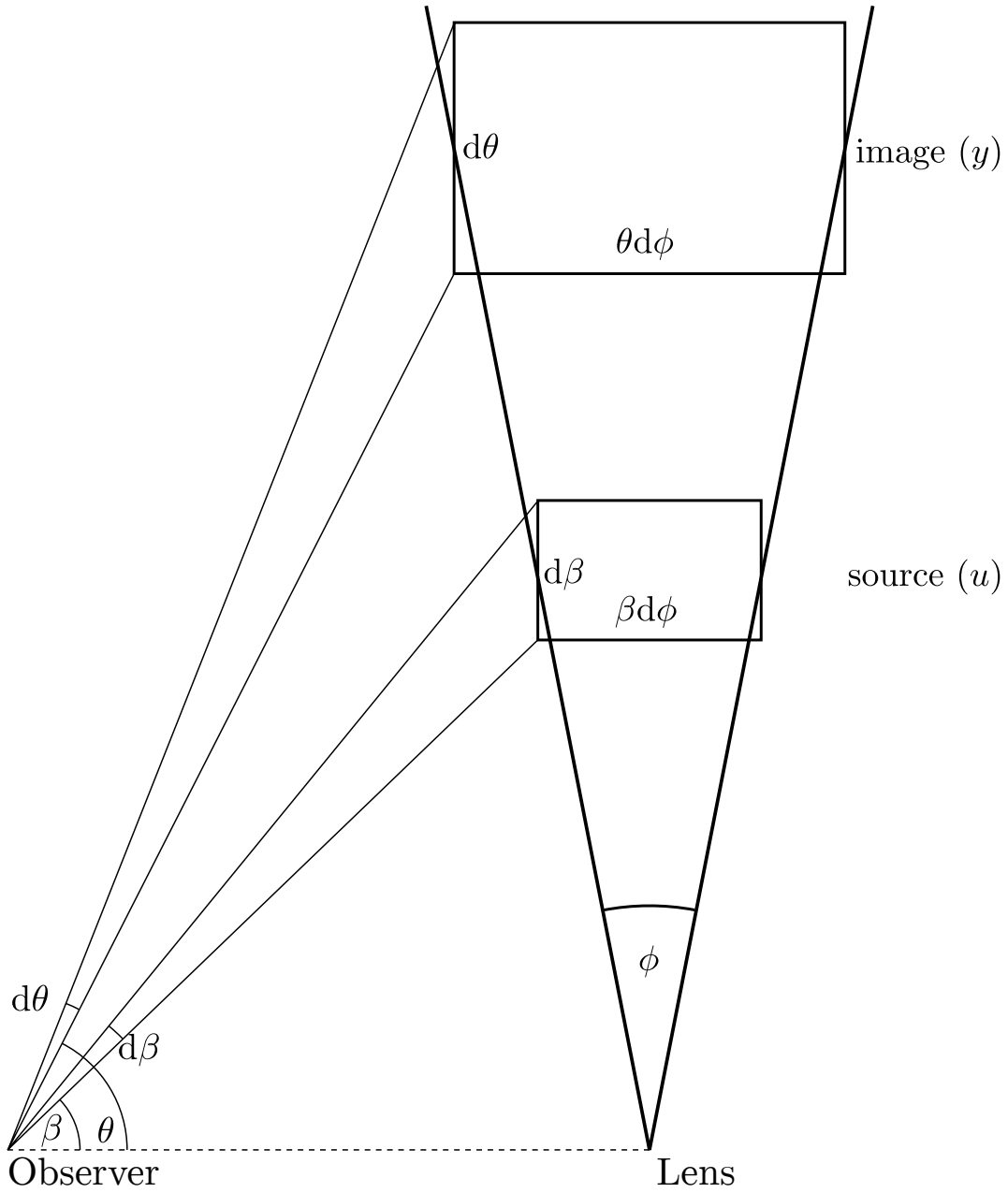


FIGURE A.1: The geometry showing the area transformation when going from a source to an image during a magnification.

The determinant gives a ratio of $\frac{\mathrm{d}A}{\mathrm{d}'A}$, however the magnification is the inverse of this, therefore

$$A = \frac{1}{detJ} = \frac{\theta}{\beta}\frac{\mathrm{d}\theta}{\mathrm{d}\beta} \ . \tag{A.5}$$

This equation is still in terms of θ and β, but with careful substitution these can be eliminated. Taking the single lens equation 2.5 and substituting in θ_E (equaiton 2.6) then equating to zero provides the following quadratic

$$\theta^2 - \theta\beta - \theta_E^2 = 0 \ . \tag{A.6}$$

This can be solved for θ as

$$\theta = \frac{1}{2}\left[\beta \pm (\beta^2 + 4\theta_E^2)^{\frac{1}{2}}\right] \ , \tag{A.7}$$

and differentiated to be

$$\frac{\mathrm{d}\theta}{\mathrm{d}\beta} = \frac{1}{2} \pm \frac{1}{2}\beta\left(\beta^2 + 4\theta_E^2\right)^{-\frac{1}{2}} \ . \tag{A.8}$$

These two terms (A.7 and A.8) can now be substituted into equation A.5

$$A = \frac{1}{2}\left[1 \pm \frac{(\beta^2 + 4\theta_E^2)^{\frac{1}{2}}}{\beta}\right]\left[1 \pm \frac{\beta}{(\beta^2 + 4\theta_E^2)^{\frac{1}{2}}}\right] \ . \tag{A.9}$$

Taking out common factors and re-arranging allows this to be reduced down to

$$A = \frac{1}{4}\left[2 \pm \left(\frac{\sqrt{u^2+4}}{u} + \frac{u}{\sqrt{u^2+4}}\right)\right] \tag{A.10}$$

$$A_\pm = \frac{1}{4}\left(2 \pm \frac{2u^2+4}{u\sqrt{u^2+4}}\right) \ . \tag{A.11}$$

The total magnification of a lensing event is the sum of the magnification of all the images created by the lens. In the case of a single lens, two images are created with opposite parity. Therefore the total magnification is the sum of these opposite parity images

$$A_{total} = |A_+| + |A_-| = \frac{1}{4}\left|2 + \frac{2u^2+4}{u\sqrt{u^2+2}}\right| + \frac{1}{4}\left|2 - \frac{2u^2+4}{u\sqrt{u^2+4}}\right| \quad . \tag{A.12}$$

This equation can be reduced further to the recognized single lens equation due to the limits of the function it contains

$$\frac{2u^2+4}{u\sqrt{u^2+4}} \quad . \tag{A.13}$$

Equation A.13 always takes the sign of u, this means that should u be negative, it can be seen that the only change to equation A.12 will be the order of the absolute values, which will result in no difference to the answer. Now consider if u is always positive, and look at the limits of equation A.13 when $u \to 0$ and $u \to \infty$.

As $u \to 0$ equation A.13 tends to ∞, and as $u \to \infty$ it tends to 2

$$\lim_{u\to 0} \frac{2u^2+4}{u\sqrt{u^2+4}} = \infty \tag{A.14}$$

$$\lim_{u\to \infty} \frac{2u^2+4}{u\sqrt{u^2+4}} = 2 \quad . \tag{A.15}$$

These could be the limits of the function, but currently the area between $u = 0$ and $u = \infty$ is unknown and the limiting values may lie in this region which has not been considered. To check for this, the first derivative test can be performed. Taking the derivative of equation A.13 will be able to show if any turning points exist, or if the chain diverges. Should any turning points exist then these need to be investigated to determine if they cause either of the limits to change. The derivative of equation A.13 is

$$\frac{d\left(\frac{2u^2+4}{u\sqrt{u^2+4}}\right)}{du} = -\frac{16}{u^2(u^2+4)^{\frac{3}{2}}} \quad . \tag{A.16}$$

In the case of a turning point in the function, the gradient of a plot would be 0, therefore any solutions will exist where equation A.16 $= 0$. Performing this test shows that there are no solutions, and therefore no turning points of the function (equation A.13). This proves that for all positive u values the lower limit of equation A.13 is 2.

With this information, it is possible to rearrange equation A.12. Using the new lower limit that equation A.13 is always > 2 allows the absolute term with the negative sign to re-arranged as shown

$$\left|2 - \frac{2u^2+4}{u\sqrt{u^2+4}}\right| \Rightarrow \frac{2u^2+4}{u\sqrt{u^2+4}} - 2 \ . \tag{A.17}$$

Substituting this into equation A.12 produces

$$A_{total} = |A_+| + |A_-| = \frac{1}{4}\left|2 + \frac{2u^2+4}{u\sqrt{u^2+2}}\right| + \frac{1}{4}\left(\frac{2u^2+4}{u\sqrt{u^2+4}} - 2\right) \ , \tag{A.18}$$

which simplifies to the recognized magnification equation for a single lens event

$$A = \frac{u^2+2}{u\sqrt{u^2+4}} \ . \tag{A.19}$$

Appendix B

Single lens source separation derivation

Figure B.1 below provides a graphical representation of the source passing behind a single lens. Using the geometry of this figure it is possible to determine an expression for u, the distance between the lens and the source in the lens plane.

Consider a source moving from left to right across the circular Einstein ring in Figure B.1. To calculate the distance between the lens and source, first the source position needs to be found. Fortunately due to the circular symmetry of a single lens, there is no need to consider the angle of the trajectory as it can always be made to be parallel to an axis. However, there are no physical distances known, only timings, so all coordinates are in units of Einstein time (t_E). Using 2-dimensional cartesian coordinates means that one dimension (u_2 in Figure B.1) is always equal to u_0, the separation at closest approach, leaving only the other dimension to be calculated. As the only known information are timings, the position along the trajectory is determined by taking its fractional distance across the Einstein ring. The time taken for the source to travel the diameter of the Einstein ring is t_E, the Einstein time. Together with t_0, the time of closest approach, a fractional distance across the Einstein ring diameter can be derived, as seen here,

$$u_1 = \frac{t - t_0}{t_E} \quad . \tag{B.1}$$

With both the u_1 and u_2 coordinates of the source, the distance to the lens can be found using Pythagorus' theorem

$$u = \sqrt{u_0^2 + \left(\frac{t - t_0}{t_E}\right)^2} \quad . \tag{B.2}$$

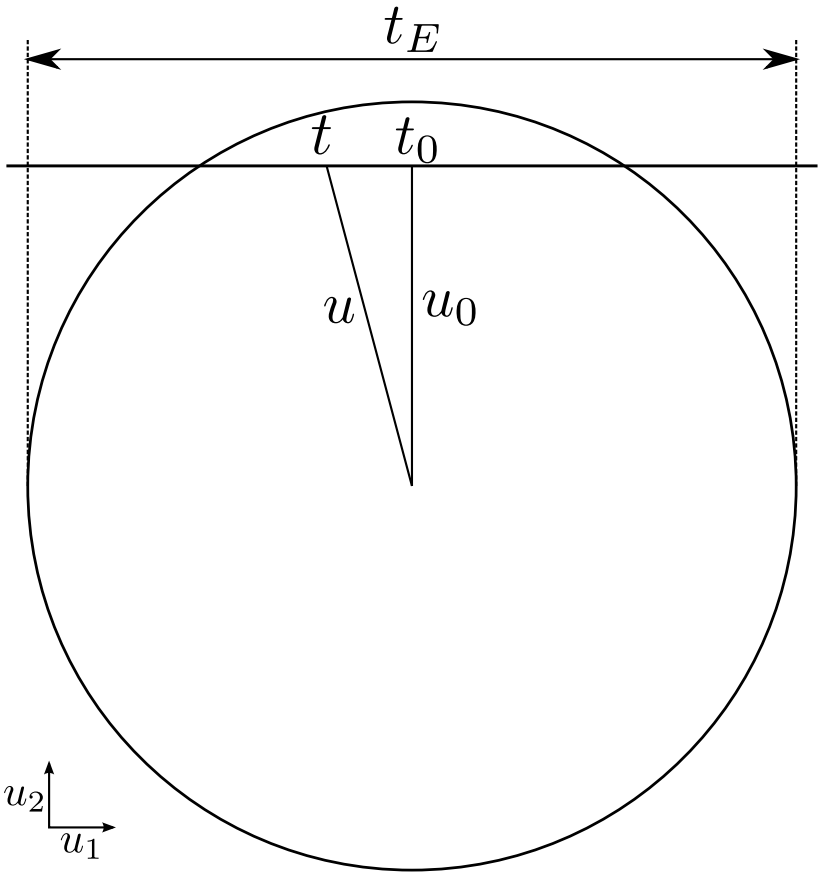


FIGURE B.1: The geometry of a single source passing behind a single lens, in the lens plane in 2-dimensional cartesian coordinates.

Appendix C

Derivation of the complex co-ordinate binary lens equation

The following method has been used to derive the complex coordinate binary lens equation used to determine the possible image positions of the lensed source.

As previously derived (equation 2.4) the lens equation can be written as

$$\beta = \theta - \sum_{i=1}^{N} \frac{4GM_i}{Dc^2} \frac{\theta - \theta_{m_i}}{|\theta - \theta_{m_i}|^2} \quad . \tag{C.1}$$

Therefore the binary lens equation (N=2) is

$$\beta = \theta - \frac{4GM_1}{Dc^2} \frac{\theta - \theta_{m_1}}{|\theta - \theta_{m_1}|^2} - \frac{4GM_2}{Dc^2} \frac{\theta - \theta_{m_2}}{|\theta - \theta_{m_2}|^2} \quad . \tag{C.2}$$

Consider the following substituations

$$\theta_E^2 = \frac{4GM}{Dc^2} \quad , \quad \varepsilon_i = \frac{M_i}{M} \tag{C.3}$$

$$\varepsilon_i \theta_E^2 = \frac{M_i}{M} \frac{4GM}{Dc^2} \Rightarrow \frac{4GM_i}{Dc^2} \tag{C.4}$$

where M is the total mass. Putting these into the binary lens equation C.2 gives

$$\beta = \theta - \varepsilon_1 \theta_E^2 \frac{\theta - \theta_{m_1}}{|\theta - \theta_{m_1}|^2} - \varepsilon_2 \theta_E^2 \frac{\theta - \theta_{m_2}}{|\theta - \theta_{m_2}|^2} \quad . \tag{C.5}$$

Scaling all angles by θ_E produces a new coordinate system for the source position (β) and image position (θ) respectivly

$$u = \frac{\beta}{\theta_E} \quad , \quad y = \frac{\theta}{\theta_E} \tag{C.6}$$

which can be made use of in the binary lens equation by dividing through by θ_E as shown

$$u = y - \varepsilon_1 \theta_E \frac{\theta - \theta_{m_1}}{|\theta - \theta_{m_1}|^2} - \varepsilon_2 \theta_E \frac{\theta - \theta_{m_2}}{|\theta - \theta_{m_2}|^2} \ . \tag{C.7}$$

Both u and y are vector positions of the source and image respectively. By considering a complex coordinate system, these can be expressed as

$$\zeta = u_1 + u_2 i \quad , \quad z = y_1 + y_2 i \ . \tag{C.8}$$

This leads to the complex coordinate binary lens equation

$$\zeta = z - \varepsilon_1 \theta_E \frac{\theta - \theta_{m_1}}{|\theta - \theta_{m_1}|^2} - \varepsilon_2 \theta_E \frac{\theta - \theta_{m_2}}{|\theta - \theta_{m_2}|^2} \ . \tag{C.9}$$

The scaled y value (C.6), re-arranged in terms of θ, $\theta = y\theta_E$, substituted into the binary lens equation (C.9) leads to a cancelation of all θ_E terms as shown

$$\zeta = z - \varepsilon_1 \frac{z - z_{m_1}}{|z - z_{m_1}|^2} - \varepsilon_2 \frac{z - z_{m_2}}{|z - z_{m_2}|^2} \ . \tag{C.10}$$

By definition the squared modulus of a complex term, $|x|^2 = x\bar{x}$, which produces

$$\zeta = z - \varepsilon_1 \frac{z - z_{m_1}}{(z - z_{m_1})(\bar{z} - \bar{z}_{m_1})} - \varepsilon_2 \frac{z - z_{m_2}}{(z - z_{m_2})(\bar{z} - \bar{z}_{m_2})} \ . \tag{C.11}$$

This reduces to the final equation, the complex coordinate binary lens equation

$$\zeta = z - \frac{\varepsilon_1}{\bar{z} - \bar{z}_{m_1}} - \frac{\varepsilon_2}{\bar{z} - \bar{z}_{m_2}} \ . \tag{C.12}$$

Applying the same algebra to the multiple lens case, produces the more general form for a complex coordinate N lens equation,

$$\zeta = z - \sum_{i=1}^{N} \frac{\varepsilon_i}{\bar{z} - \bar{z}_{m_i}} \quad . \tag{C.13}$$

Appendix D

Solutions to the 5^{th} order polynomial binary lens equation

Starting with the complex coordinate binary lensing equation derived in Appendix C and substituting in its complex conjugate re-arranged for $\bar{z}$, removes all $\bar{z}$'s, allowing the equation to be solved for z. This will be shown in the following steps

$$\zeta = z - \frac{\varepsilon_1}{\bar{z} - \bar{z}_{m_1}} - \frac{\varepsilon_2}{\bar{z} - \bar{z}_{m_2}} \tag{D.1}$$

$$\bar{\zeta} = \bar{z} - \frac{\varepsilon_1}{z - z_{m_1}} - \frac{\varepsilon_2}{z - z_{m_2}} \ . \tag{D.2}$$

By defining your coordinate system as having the origin mid way between the two lens objects ($z_{m_1} = a, z_{m_2} = -a$) as well as considering the mass ratios arranged so ε_2 can be expressed in terms of ε_1 ($\varepsilon_2 = 1 - \varepsilon_1$),

$$\zeta = z - \frac{\varepsilon_1}{\bar{z} - a} - \frac{1 - \varepsilon_1}{\bar{z} + a} \tag{D.3}$$

$$\bar{\zeta} = \bar{z} - \frac{\varepsilon_1}{z - a} - \frac{1 - \varepsilon_1}{z + a} \ . \tag{D.4}$$

Substituting D.4 re-arranged for $\bar{z}$ into D.3 results in an equation for ζ in terms of z, a, ε_1, and $\bar{\zeta}$

$$\begin{aligned}\zeta \quad = z \quad & -\frac{\varepsilon_1(-z+a)(z+a)}{-2\varepsilon_1 a - z + a - \bar{\zeta}z^2 + \bar{\zeta}a^2 + az^2 - a^3} \\ & +\frac{(-1+\varepsilon_1)(-z+a)(z+a)}{-2\varepsilon_1 a - z + a - \bar{\zeta}z^2 + \bar{\zeta}a^2 - az^2 + a^3} \quad .\end{aligned} \tag{D.5}$$

Lastly, multiplying by the denominator results in a 5^{th} order complex polynomial for z with coefficients

$$z^5 : \bar{\zeta}^2 - a^2 \tag{D.6}$$

$$z^4 : \bar{\zeta} + a - 2\varepsilon_1 a - \zeta\bar{\zeta}^2 + \zeta a^2 \tag{D.7}$$

$$z^3 : -2a\bar{\zeta} + 2a^4 - 2\zeta\bar{\zeta} + 4\varepsilon_1 a\bar{\zeta} - 2\bar{\zeta}^2 a^2 \tag{D.8}$$

$$z^2 : -2\zeta a^4 - 4\zeta\varepsilon_1 a\bar{\zeta} + 4a^3\varepsilon_1 + 2\zeta\bar{\zeta}^2 a^2 - \zeta - 2a^3 - a + 2\varepsilon_1 a + 2\zeta a\bar{\zeta} \tag{D.9}$$

$$z^1 : 2\zeta\bar{\zeta}a^2 + \bar{\zeta}^2 a^4 + 2\bar{\zeta}a^3 - a^6 + 2a^2 - 4\varepsilon_1 a^2 + 4\varepsilon_1^2 a^2 - 4\zeta\varepsilon_1 a + 2\zeta a - 4\varepsilon_1 a^3\bar{\zeta} \tag{D.10}$$

$$z^0 : -2\varepsilon_1 a^5 - a^3 + a^5 - 4\zeta\varepsilon_1^2 a^2 + 2a^3\varepsilon_1 + 4\zeta a^3\bar{\zeta} + 4\zeta\varepsilon_1 a^2 - \zeta\bar{\zeta}a^4 - 2\zeta\bar{\zeta}a^3 - \zeta a^2 + \zeta a^6 - a^4\bar{\zeta} \quad . \tag{D.11}$$

This equation cannot be solved analytically, so a numerical solution is required. The five solutions correspond to the possible image positions from the lens. These solutions are then required to be put back into the original binary lens equation D.1 to test if they are true solutions.

Appendix E

Binary lens magnification derivation

The magnification of a binary lens can be found in a similar way to that of a single lens. As shown in Appendix A, the magnification $A = \frac{1}{detJ}$, where the Jacobian matrix of a multiple lens system can be expressed using

$$\zeta = z - \frac{\varepsilon_1}{\overline{z} - \overline{z}_{m_1}} - \frac{\varepsilon_2}{\overline{z} - \overline{z}_{m_2}} \tag{E.1}$$

$$\overline{\zeta} = \overline{z} - \frac{\varepsilon_1}{z - z_{m_1}} - \frac{\varepsilon_2}{z - z_{m_2}} \ , \tag{E.2}$$

to be

$$J = \begin{bmatrix} \frac{\partial \zeta}{\partial z} & \frac{\partial \zeta}{\partial \overline{z}} \\ \frac{\partial \overline{\zeta}}{\partial z} & \frac{\partial \overline{\zeta}}{\partial \overline{z}} \ , \end{bmatrix} \tag{E.3}$$

where the determinant is

$$detJ = \frac{\partial \zeta}{\partial z}\frac{\partial \overline{\zeta}}{\partial \overline{z}} - \frac{\partial \zeta}{\partial \overline{z}}\frac{\partial \overline{\zeta}}{\partial z} \ . \tag{E.4}$$

As $\overline{\frac{\partial \overline{a}}{\partial b}} = \frac{\partial a}{\partial b}$ and $a\overline{a} = |a|^2$,

$$detJ = \left|\frac{\partial \zeta}{\partial z}\right|^2 - \left|\frac{\partial \zeta}{\partial \overline{z}}\right|^2 \ . \tag{E.5}$$

From Equation E.1, the partial derivatives of equation E.5 are found to be $\frac{\partial \zeta}{\partial z} = 1$ and $\frac{\partial \zeta}{\partial \overline{z}} = \sum_{i=0}^{N} \frac{\epsilon_i}{(\overline{z}-\overline{z}_{m,i})^2}$ resulting in the determinant of the Jacobian being,

$$detJ = 1 - \left| \sum_{i=0}^{N} \frac{\epsilon_i}{(\overline{z} - \overline{z}_{m,i})^2} \right|^2 \quad . \tag{E.6}$$

Appendix F

Binary lens coordinate system derivation

The coordinate system for a binary lens is the same as for a single lens, but circular symmetry no longer exists. This means that using only the distance between the source and lens is not sufficient and instead a two dimensional coordinate system needs to be used, u_1 and u_2. As with the single lens, these coordinates are in units of angular Einstein radius and their values can be found in a similar way. The coefficients for solving the 5^{th} order polynomial have been derived with the origin of the binary lens being exactly half way between the two lens masses, and the u_1 axis to be along the line connecting m_1 and m_2. Figure F.1 shows the geometry of a binary lens system.

The position along the trajectory is determined in the same way as for the single lens, but this time the angle of the trajectory is important. If the source passes through the origin ($u_{mp} = 0$), which is defined as the mid-point between the two masses, the source's position along the trajectory is determined by $d_{mp} = \left(\frac{t-t_{mp}}{t_E}\right)$, where t_{mp} is the time when the source is at the mid-point between the masses. Using simple trigonometry the coordinates are determined to be

$$u_1 = \left(\frac{t - t_{mp}}{t_E}\right) \cos\phi \ , \ \ u_2 = \left(\frac{t - t_{mp}}{t_E}\right) \sin\phi \ . \tag{F.1}$$

These equations describe the source passing through the mid-point ($u_{mp} = 0$), however in the majority of cases this is not true. From the geometry in Figure F.1, modifications to equations F.1 are found to account for a trajectory that does not pass through the mid-point. With trigonometry, the coordinates of the point on the trajectory at t_{mp} are found and added to equation F.1, which results in the the source coordinates in a binary lens plane

$$u_1 = \left(\frac{t - t_{mp}}{t_E}\right)\cos\phi - u_{mp}\sin\phi \ , \quad u_2 = \left(\frac{t - t_{mp}}{t_E}\right)\sin\phi + u_{mp}\cos\phi \ . \tag{F.2}$$

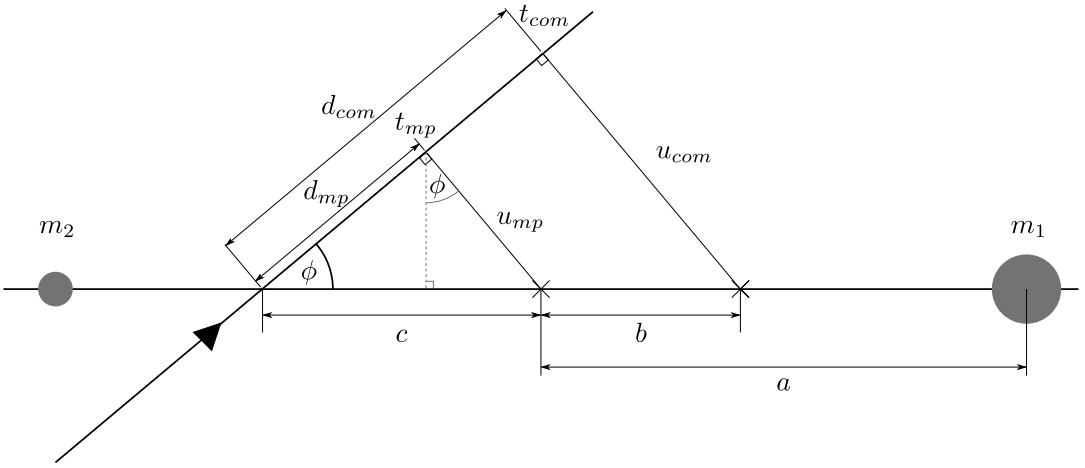


FIGURE F.1: The geometry of a binary lens system, showing the two lens masses and a trajectory passing through at an angle ϕ a with a distance of closest approach to the mid way point between the masses of u_{mp}.

In the microlensing community u_0 is typically quoted as the distance of closest approach to the centre of mass of a system (u_{com}), not the mid-point between lens masses (u_{mp}). A conversion between these two terms is therefore required. From Figure F.1 the length u_{com} can be described as

$$u_{com} = (c + b)\sin\phi \ , \tag{F.3}$$

where ϕ is the angle of the trajectory, b the distance between the centre of mass and the mid way point and c the distance between the axis crossing point and the mid-point,

$$c = \frac{u_{mp}}{\sin\phi} \ . \tag{F.4}$$

Substituting F.4 into equation F.3 provides a transformation for the coordinates of closest approach at the mid-point to those at the centre of mass

$$u_{com} = u_{mp} + b\sin\phi \ . \tag{F.5}$$

A similar transformation needs to be made for the time of closest approach t_{com}. A new time of closest approach means that the source has travelled a different fractional distance along the trajectory, found by $d_{com} - d_{mp}$. As all the dimensions in this geometry

are in units of Einstein radii, and the time for the source to travel one Einstein radius is defined as t_E, the source velocity is determined to be $\frac{1}{t_E}$. With a velocity and distance between the two defined points of closest approach, the time taken to travel the change in position is found. This time combined with the time of closest approach to the mid-point describes the desired time of closest approach to the centre of mass

$$t_{com} = t_{mp} + \frac{d_{com} - d_{mp}}{\frac{1}{t_E}} \quad . \tag{F.6}$$

Distances d_{mp} and d_{com} are not known, but they can be expressed in terms of the angle of the trajectory and the distance of closest approach

$$d_{mp} = \frac{u_{mp}}{\tan\phi} \quad , \quad d_{com} = \frac{u_{com}}{\tan\phi} \tag{F.7}$$

which by substitution results in

$$t_{com} = t_{mp} + t_E \left(\frac{u_{com}}{\tan\phi} - \frac{u_{mp}}{\tan\phi} \right) \quad . \tag{F.8}$$

Equations F.5 and F.8 are essential for converting between the commonly used closest approach to the centre of mass, and the modeller required closest approach to the mid-point.

Appendix G

Flow charts detailing the binary lens modelling process

G.1 Flow charts

This appendix includes a set of flow charts detailing the process involved in the modelling of binary lens events.

G.1.1 Making a magnification map

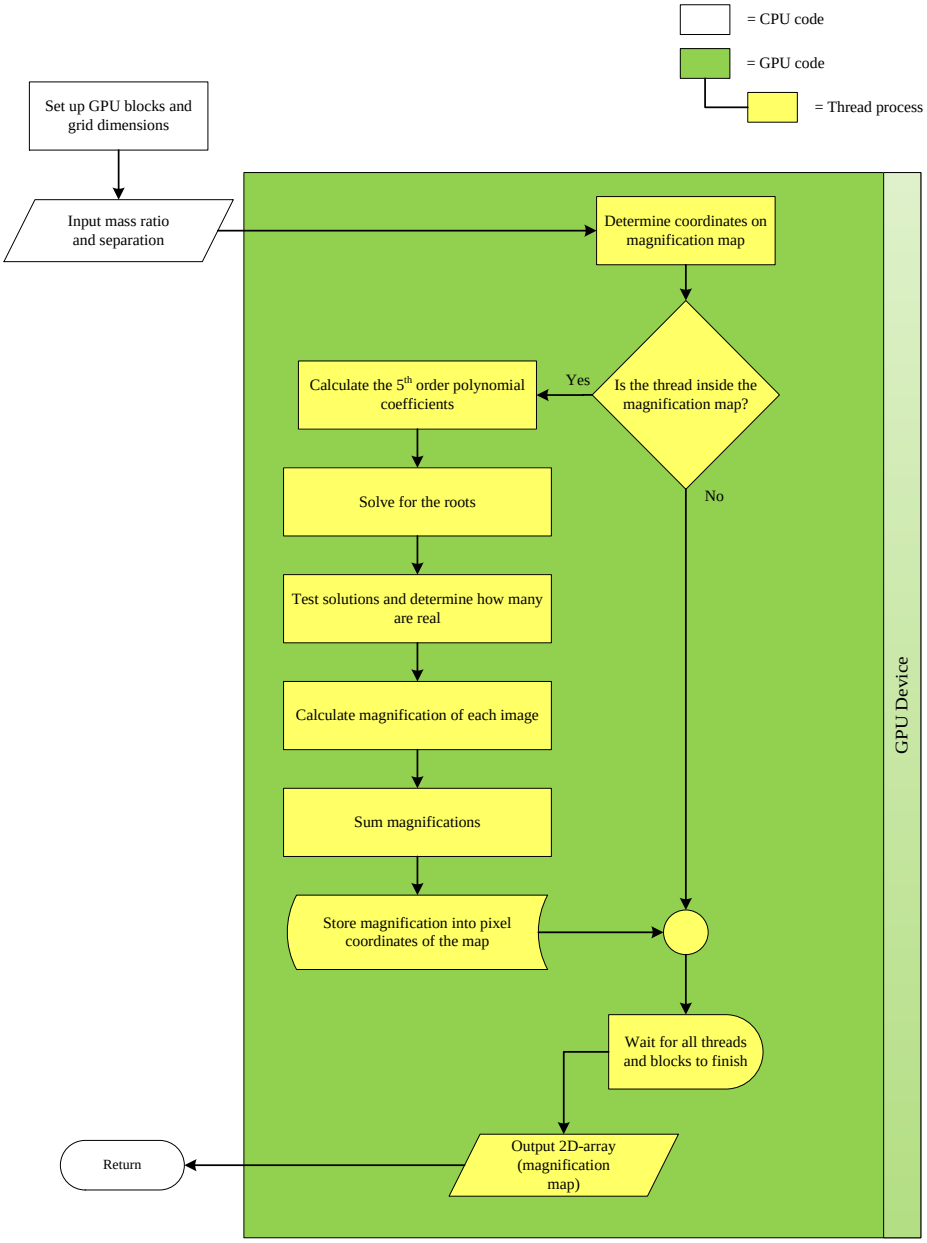


FIGURE G.1: Flow chat showing the procedure involved to make a magnification map.

G.1.2 Making a high resolution magnification map

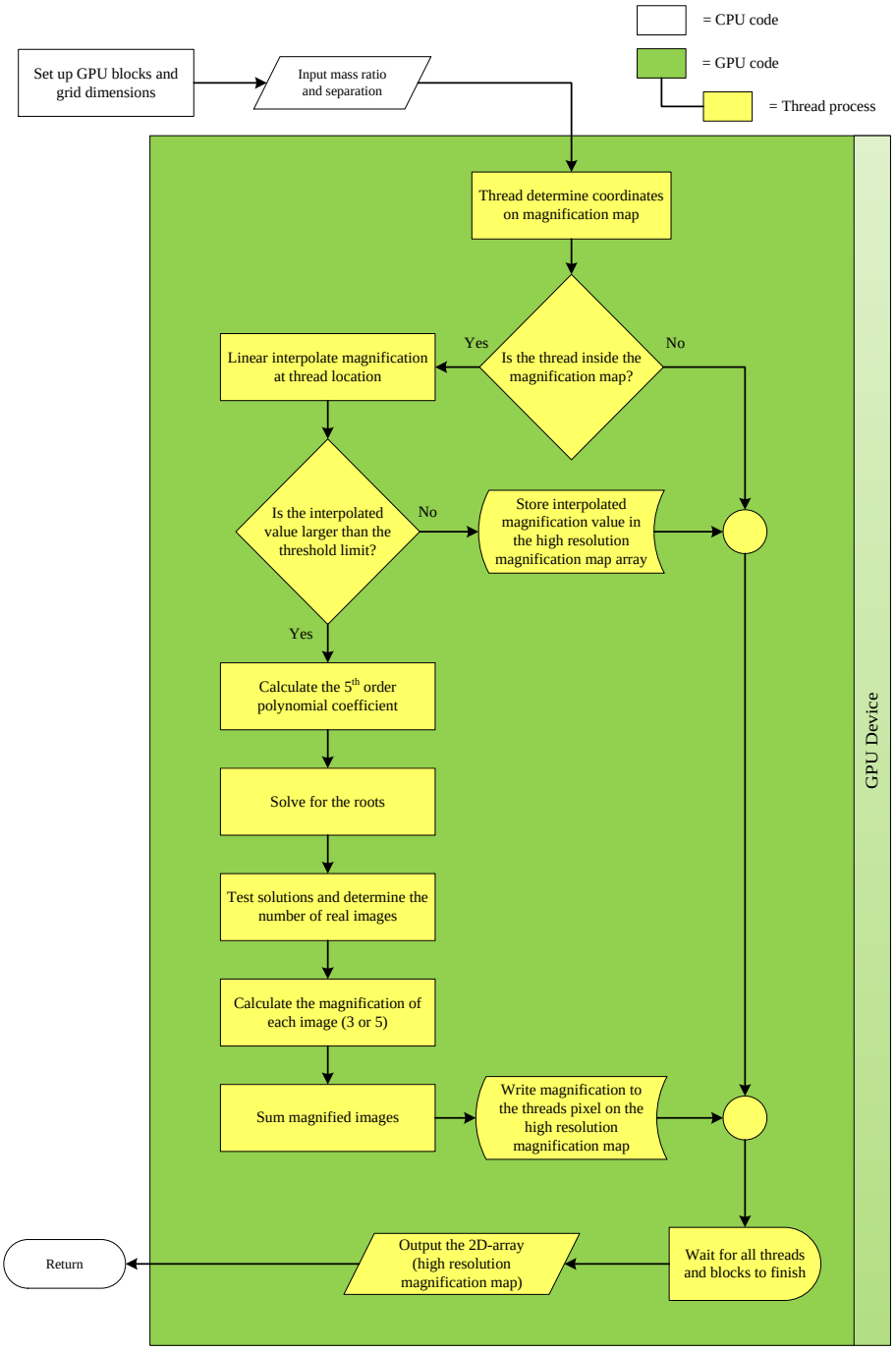


FIGURE G.2: Flow chat showing the procedure involved in making a high resolution magnification map.

G.1.3 High magnification filter

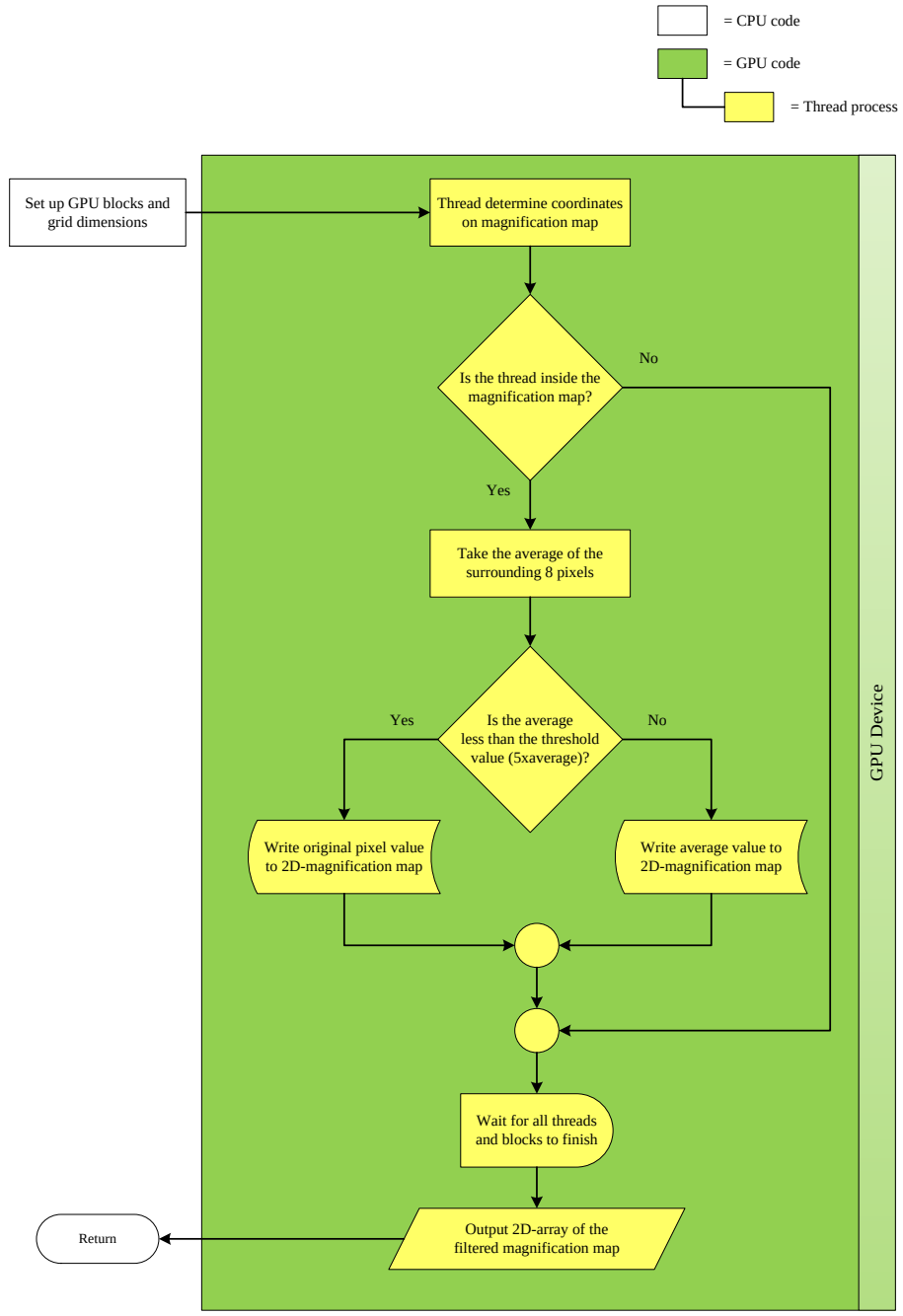


FIGURE G.3: Flow chat showing the procedure involved in filtering out high magnification pixels.

G.1.4 Convolving a magnification map

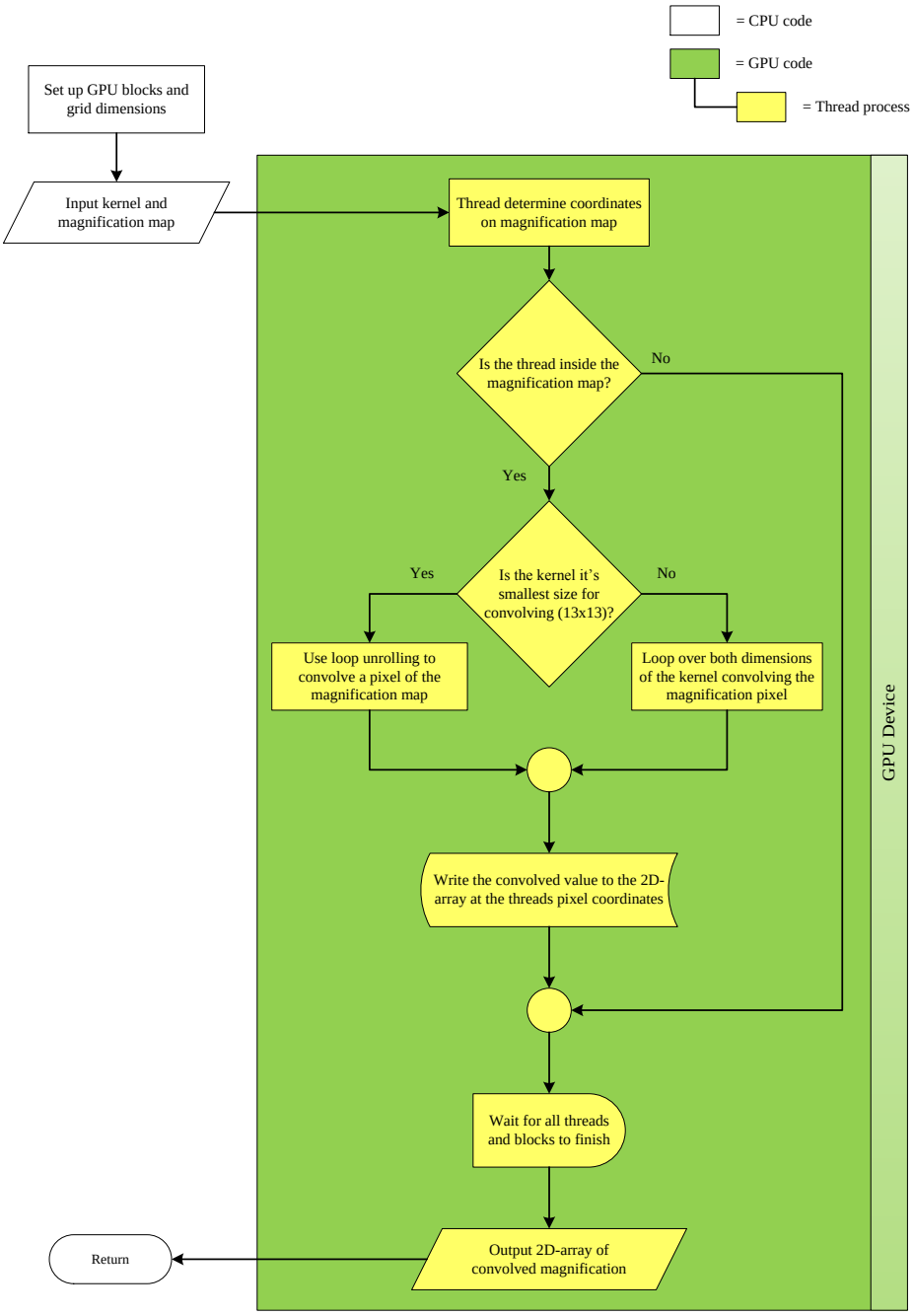


FIGURE G.4: Flow chat showing the procedure involved in convolving a magnification map.

G.1.5 Creating a set of convolution maps

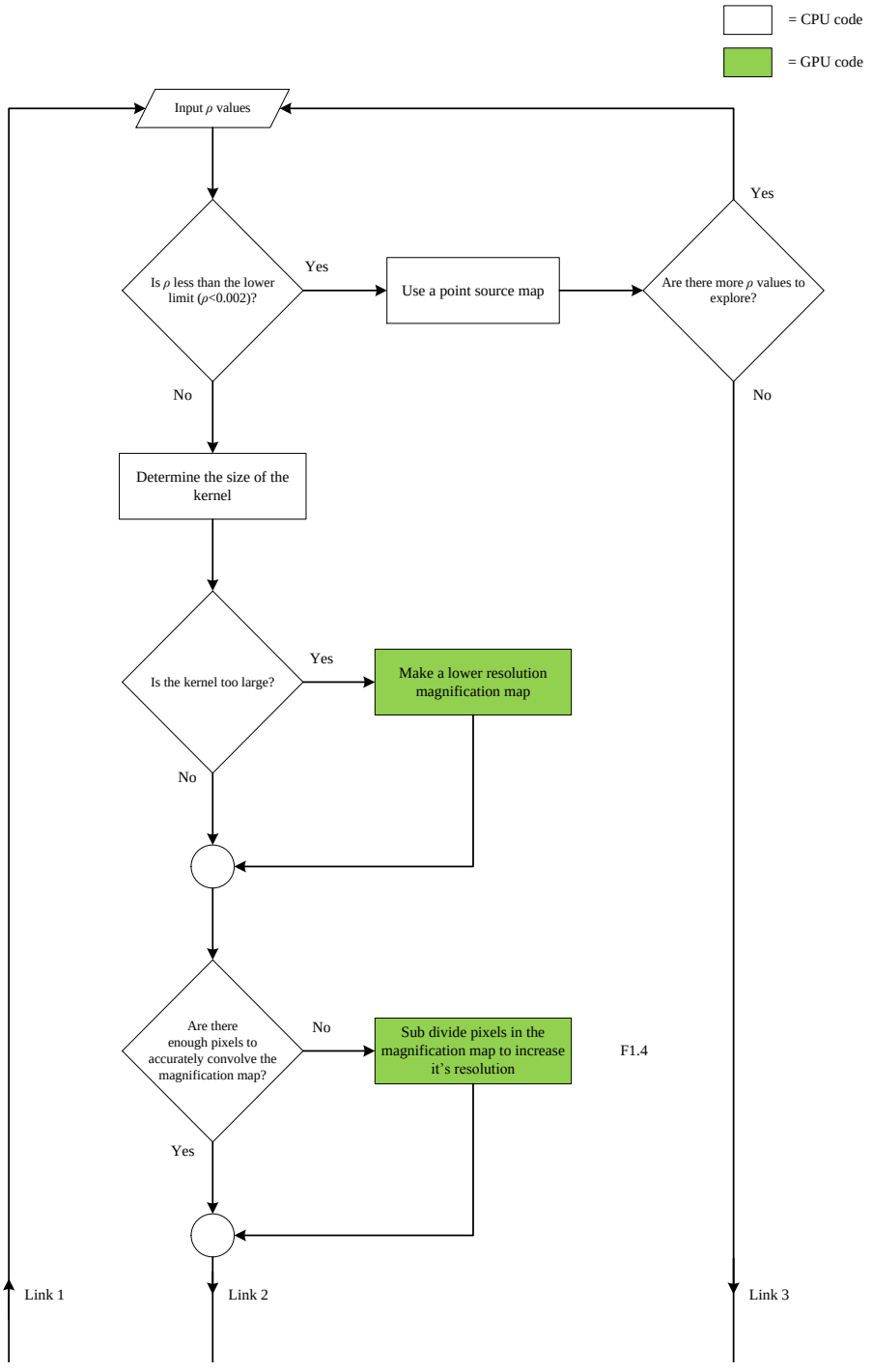


FIGURE G.5: Flow chat showing the procedure involved to create a set of convolution maps.

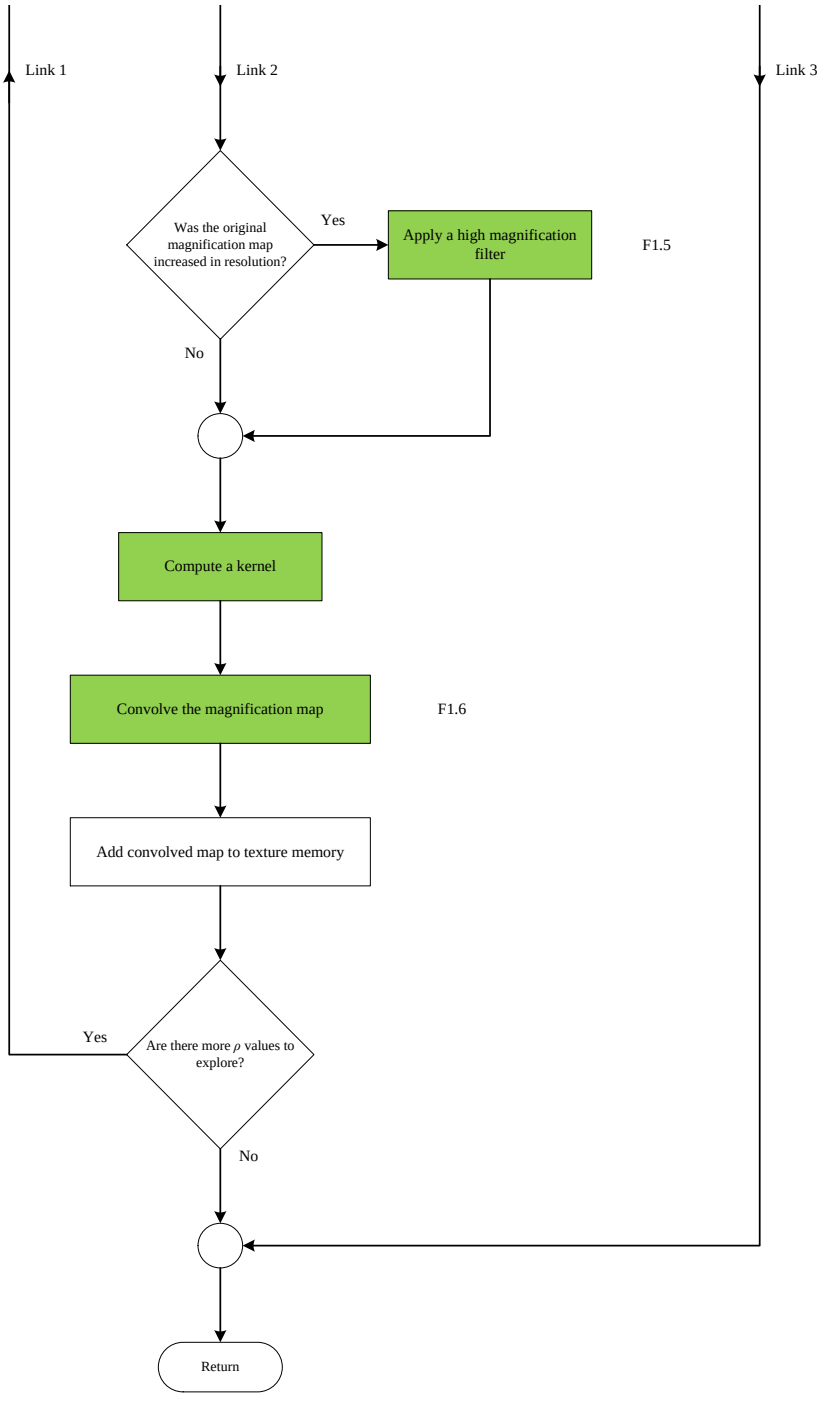


FIGURE G.6: Flow chat showing the procedure involved to create a set of convolution maps.

G.1.6 Binary model search overview

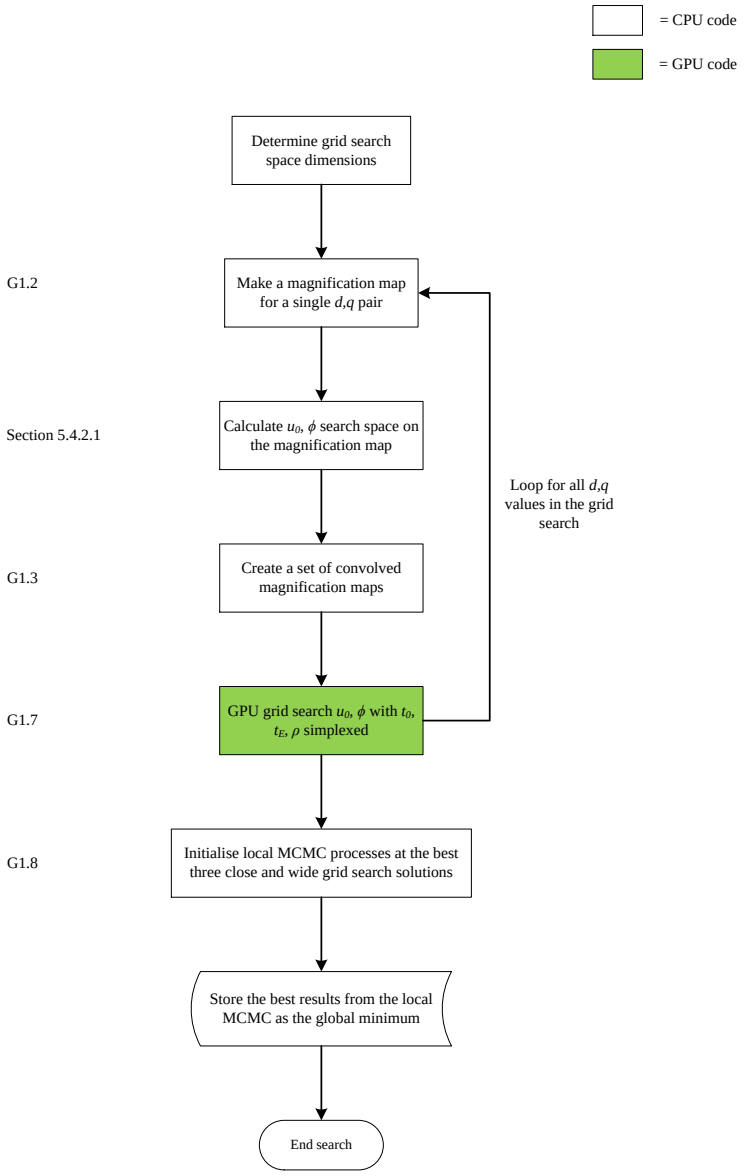


FIGURE G.7: Flow chat showing an overview of the binary lens modelling procedure.

G.1.7 Local area Markov Chain Monte Carlo search

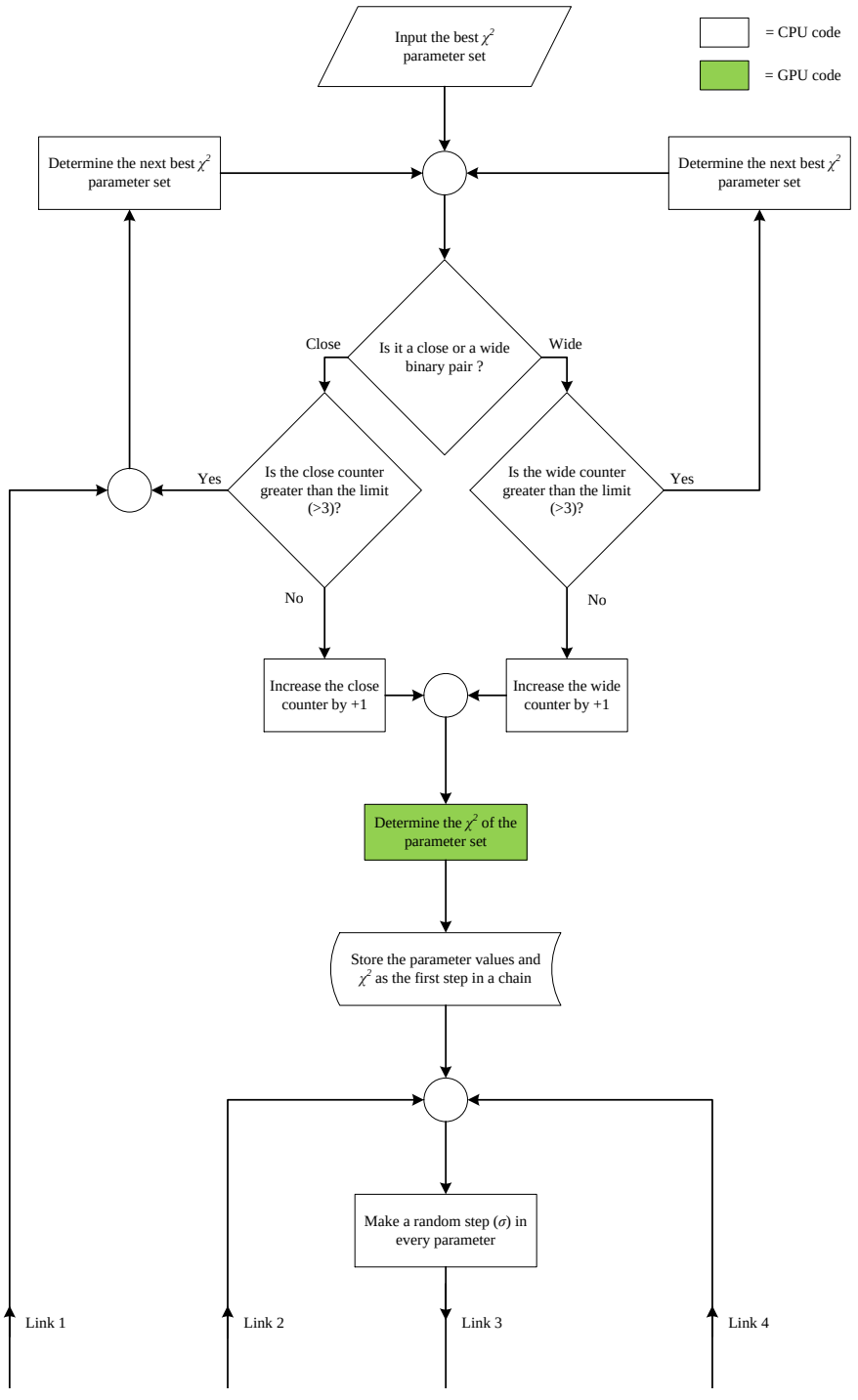


FIGURE G.8: Flow chart detailing the local MCMC procedure part 1.

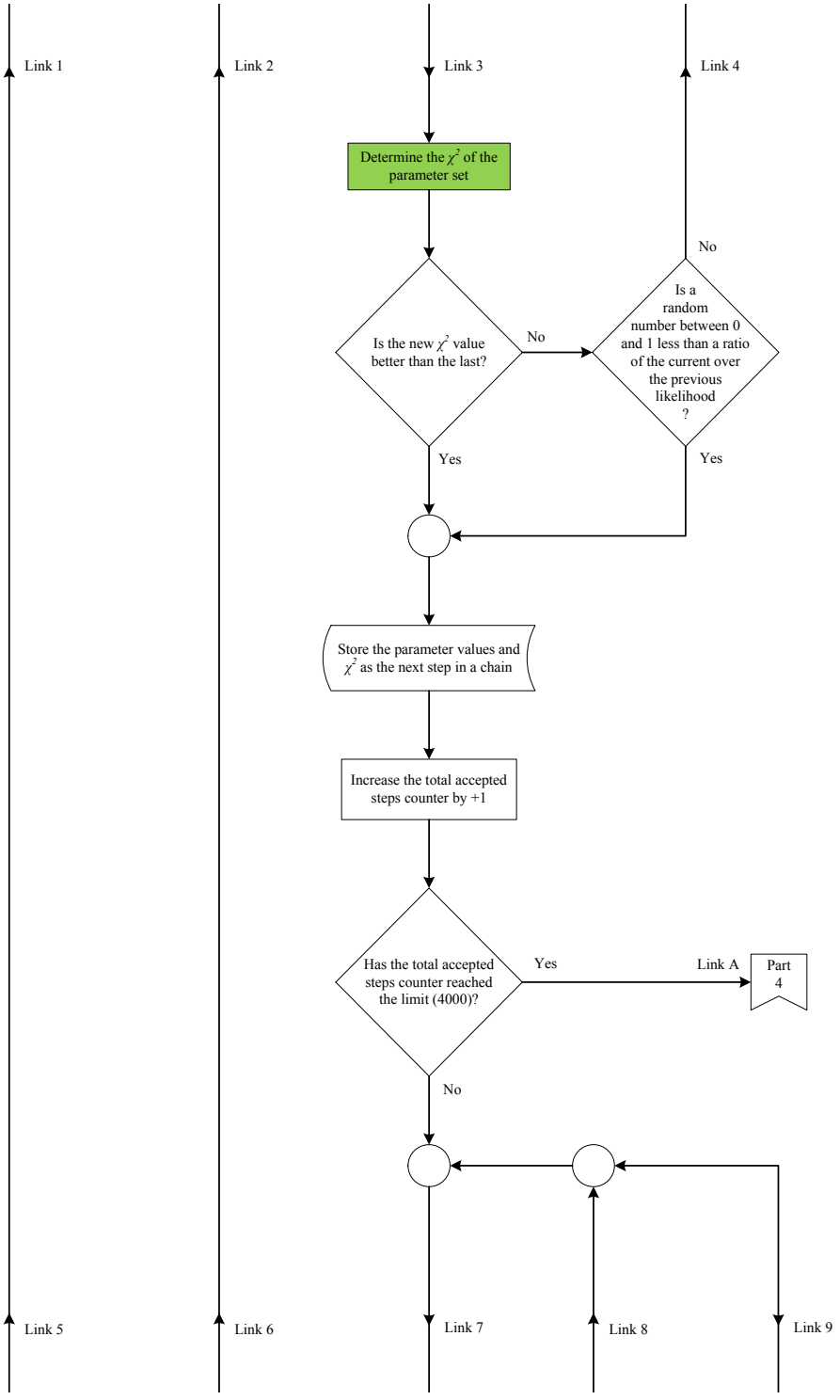


FIGURE G.9: Flow chart detailing the local MCMC procedure part 2.

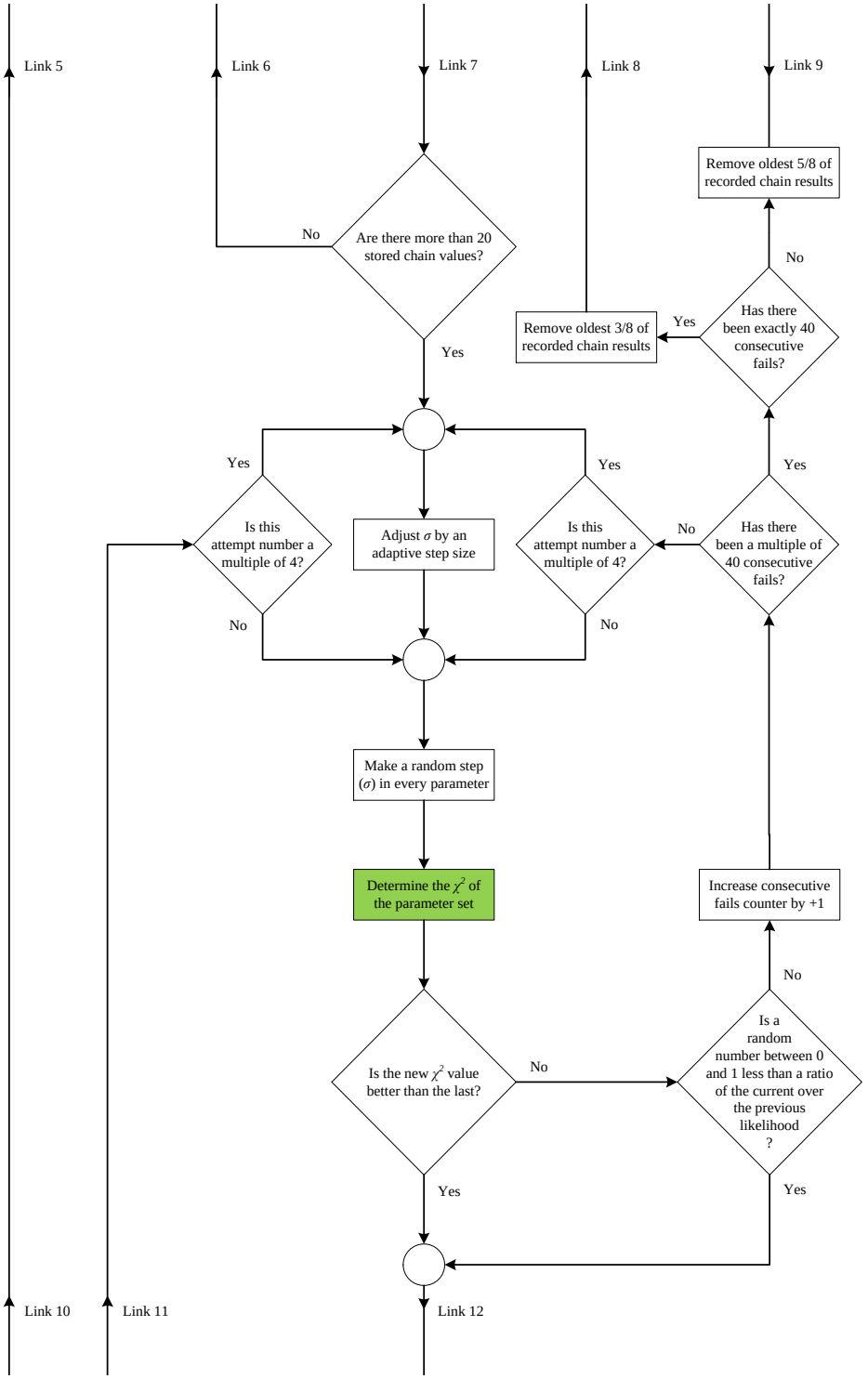


FIGURE G.10: Flow chart detailing the local MCMC procedure part 3.

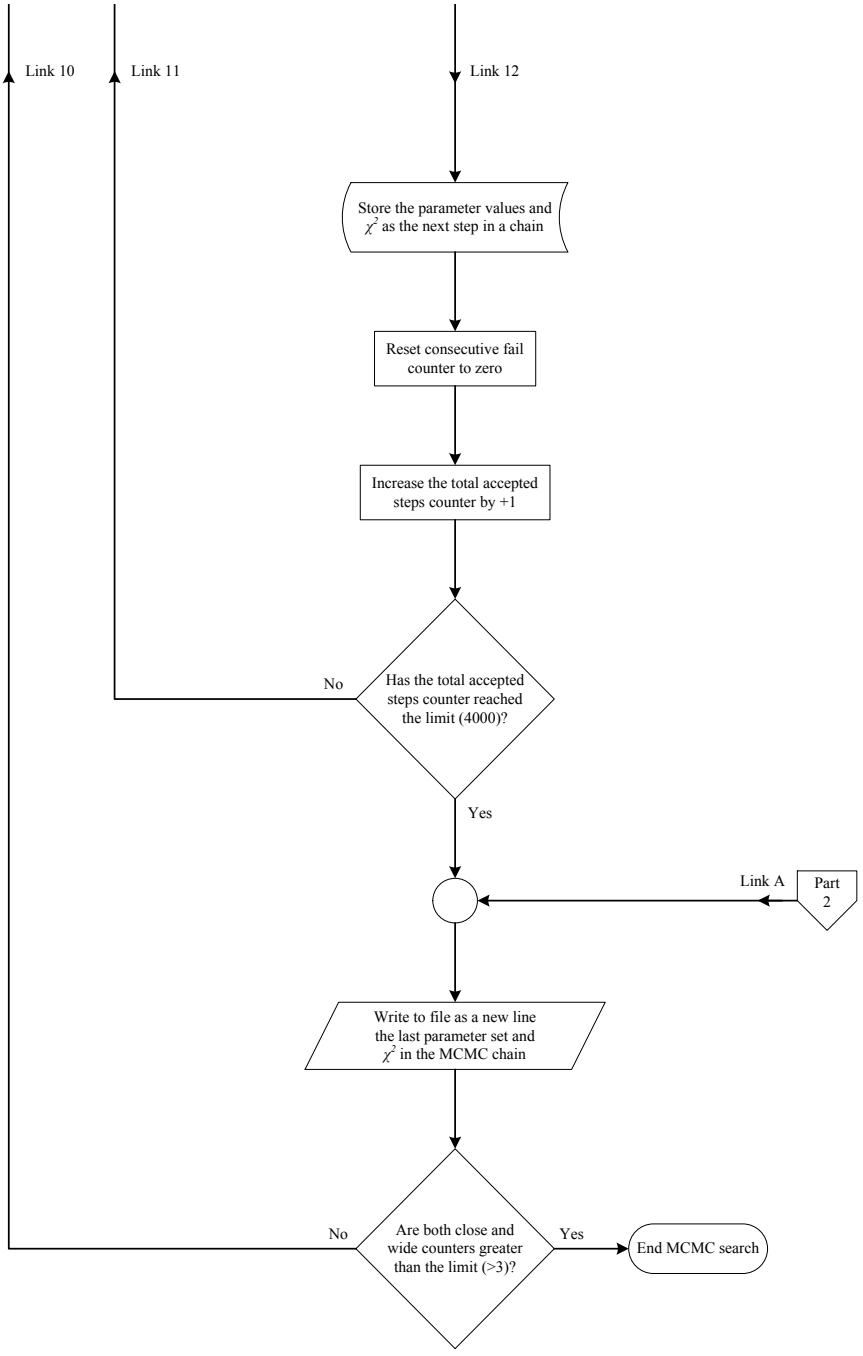


FIGURE G.11: Flow chart detailing the local MCMC procedure part 4.

G.1.8 Graphics Processing Unit u_0, ϕ grid search

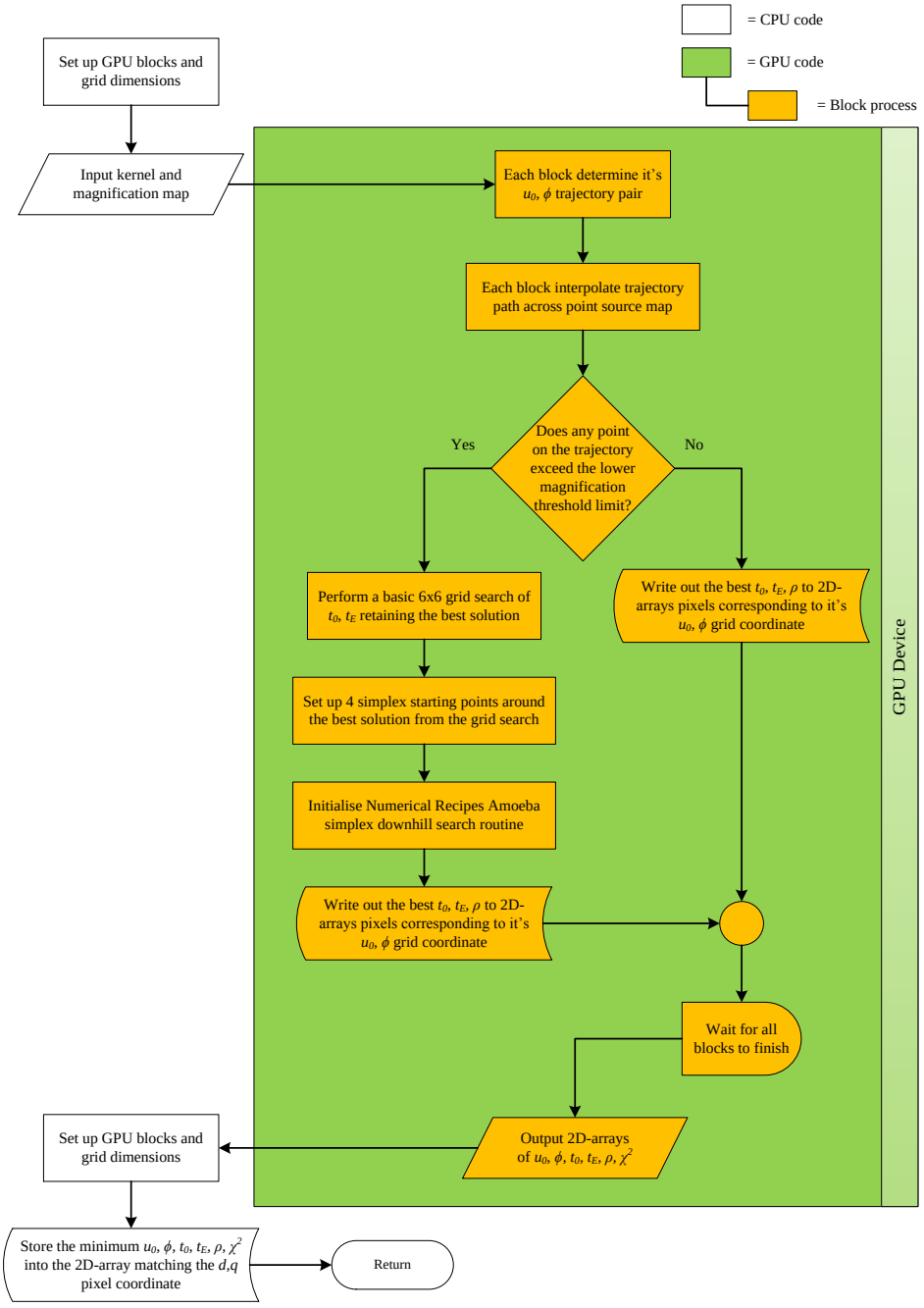


FIGURE G.12: Flow chat showing the procedure to perform a grid search on the GPU of u_0, ϕ parameter space.

Appendix H

Graphics Processing Unit device specifications

The following details are the output specifications of the GPU device, an Nvidia Tesla C2075.

===Attributes for device 0
MAX THREADS PER BLOCK:1024
MAX BLOCK DIM X:1024
MAX BLOCK DIM Y:1024
MAX BLOCK DIM Z:64
MAX GRID DIM X:65535
MAX GRID DIM Y:65535
MAX GRID DIM Z:65535
MAX SHARED MEMORY PER BLOCK:49152
TOTAL CONSTANT MEMORY:65536
WARP SIZE:32
MAX PITCH:2147483647
MAX REGISTERS PER BLOCK:32768
CLOCK RATE:1147000
TEXTURE ALIGNMENT:512
GPU OVERLAP:1
MULTIPROCESSOR COUNT:14
KERNEL EXEC TIMEOUT:0
INTEGRATED:0
CAN MAP HOST MEMORY:1
COMPUTE MODE:DEFAULT

MAXIMUM TEXTURE1D WIDTH:65536
MAXIMUM TEXTURE2D WIDTH:65536
MAXIMUM TEXTURE2D HEIGHT:65535
MAXIMUM TEXTURE3D WIDTH:2048
MAXIMUM TEXTURE3D HEIGHT:2048
MAXIMUM TEXTURE3D DEPTH:2048
MAXIMUM TEXTURE2D ARRAY WIDTH:16384
MAXIMUM TEXTURE2D ARRAY HEIGHT:16384
MAXIMUM TEXTURE2D ARRAY NUMSLICES:2048
SURFACE ALIGNMENT:512
CONCURRENT KERNELS:1
ECC ENABLED:1
PCI BUS ID:1
PCI DEVICE ID:0
TCC DRIVER:0
MEMORY CLOCK RATE:1566000
GLOBAL MEMORY BUS WIDTH:384
L2 CACHE SIZE:786432
MAX THREADS PER MULTIPROCESSOR:1536
ASYNC ENGINE COUNT:2
UNIFIED ADDRESSING:1
MAXIMUM TEXTURE1D LAYERED WIDTH:16384
MAXIMUM TEXTURE1D LAYERED LAYERS:2048

References

Abe, F., et al. 1996, in 12^{th} IAP - Variable stars and the astrophysical returns of microlensing surveys, ed. R. Ferlet, J. P. Maillard, & B. Raban

Addison, B. C., Tinney, C. G., Wright, D. J., et al. 2014, ArXiv e-prints, arXiv:1403.0652

Alard, C. 1996, in International Astronomical Union / Union Astronomique Internationale, Vol. 173, Astrophysical Applications of Gravitational Lensing, ed. C. Kochanek & J. Hewitt (Springer Netherlands), 215–220

Albrow, M. D., et al. 1996, in 12^{th} IAP - Variable stars and the astrophysical returns of microlensing surveys, ed. R. Ferlet, J. P. Maillard, & B. Raban

Albrow, M. D., Beaulieu, J.-P., Birch, P., et al. 1998, ApJ, 509, 687

Albrow, M. D., Beaulieu, J.-P., Caldwell, J. A. R., et al. 2000, ApJ, 534, 894

Alcock, C., Akerlof, C. W., Allsman, R. A., et al. 1993, Nature, 365, 621

Alcock, C., Allsman, R. A., Alves, D., et al. 1996, ApJ, 463, L67

Anderson, J. A., Lorenz, C. D., & Travesset, A. 2008, Journal of Computational Physics, 227, 5342

Aubourg, A., et al. 1993, Nature, 365, 623

Barsdell, B. R., Barnes, D. G., & Fluke, C. J. 2010, MNRAS, 408, 1936

Bate, N., Fluke, C., Barsdell, B., Garsden, H., & Lewis, G. 2010, New Astronomy

Batista, V., Gould, A., Dieters, S., et al. 2011, A&A, 529, A102

Batista, V., Beaulieu, J.-P., Gould, A., et al. 2014, ApJ, 780, 54

Beaulieu, J.-P., Albrow, M., Bennett, D., et al. 2007, The Messenger, 128, 33

Bennett, D. P. 1998, Physics Reports, 307, 97

—. 2010, ApJ, 716, 1408

Bennett, D. P., & Rhie, S. H. 1996, ApJ, 472, 660

Bennett, D. P., Batista, V., Bond, I. A., et al. 2014, ApJ, 785, 155

Bergmann, C., Endl, M., Hearnshaw, J. B., Wittenmyer, R. A., & Wright, D. J. 2014, IJA, doi:http://dx.doi.org/10.1017/S1473550414000317

Berry, R., & Burnell, J. 2006, The Handbook of Astronomical Image Processing, 2nd edn. (Willmann-Bell, Inc.)

Bode, M. F., Cardiff U. Collaboration, U. Hertfordshire Collaboration, et al. 2004, in Bulletin of the American Astronomical Society, Vol. 36, American Astronomical Society Meeting Abstracts, 1400

Bodenheimer, P., & Pollack, J. B. 1986, Icarus, 67, 391

Boettcher, E., Willman, B., Fadely, R., et al. 2013, The Astronomical Journal, 146, 94

Bond, I. A., Abe, F., Dodd, R. J., et al. 2001, MNRAS, 327, 868

Bond, I. A., Udalski, A., Jaroszyński, M., et al. 2004, ApJ, 606, L155

Boss, A. P. 1997, Science, 276, 1836

—. 2001, ApJ, 563, 367

Bozza, V. 2010, MNRAS, 408, 2188

Bozza, V., Dominik, M., Rattenbury, N. J., et al. 2012, MNRAS, 424, 902

Campbell, B., Walker, G. A. H., & Yang, S. 1988, ApJ, 331, 902

Cassan, A., Kubas, D., Beaulieu, J.-P., et al. 2012, Nature, 481, 167

Christen, J., & Fox, C. 2007, A General Purpose Scale-Independent MCMC Algorithm, Tech. Rep. I-07-16, CIMAT

Claret, A. 2000, A&A, 363, 1081

Claret, A., & Gimenez, A. 1990, A&A, 230, 412

D. Gamerman, H. L. 2006, Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference (Chapman & Hall)

Debes, J. H., & Sigurdsson, S. 2007, ApJ, 668, L167

Desidera, S., Bonomo, A. S., Claudi, R. U., et al. 2014, A&A, 567, L6

Diaz-Cordoves, J., & Gimenez, A. 1992, A&A, 259, 227

Dominik, M. 1993, in Liege International Astrophysical Colloquia, Vol. 31, Liege International Astrophysical Colloquia, ed. J. Surdej, D. Fraipont-Caro, E. Gosset, S. Refsdal, & M. Remy , 541

Dominik, M. 1995, A&AS, 109, 597

—. 1996, PhD thesis, Universität Dortmund

—. 1998a, A&A, 333, L79

—. 1998b, A&A, 329, 361

—. 2007, MNRAS, 377, 1679

Dominik, M. 2008, in Proceedings of the Manchester Microlensing Conference: The 12th International Conference and ANGLES Microlensing Workshop, ed. N. R. E. Kerins, S. Mao & L. Wyrzykowski

Dominik, M., Horne, K., Allan, A., et al. 2008, Astronomische Nachrichten, 329, 248

Dominik, M., Jørgensen, U. G., Rattenbury, N. J., et al. 2010, Astronomische Nachrichten, 331, 671

Dong, S., DePoy, D. L., Gaudi, B. S., et al. 2006, ApJ, 642, 842

Doran, M., & Müller, C. M. 2004, JCAP, 9, 3

Dumusque, X., Pepe, F., Lovis, C., et al. 2012, Nature, 491, 207

Durisen, R. H., Boss, A. P., Mayer, L., et al. 2007, Protostars and Planets V, 607

Einstein, A. 1915, Sitzungsber. preuss.Akad. Wiss., vol. 47, No.2, pp. 831-839, 1915, 47, 831

—. 1936, Science, 84, 506

—. 1936b, letter to J. Cattell, dated 18 December 1936 (Einstein Archives call no. 65-603)

Endl, M., Caldwell, D. A., Barclay, T., et al. 2014a, ApJ, 795, 151

Endl, M., Bergmann, C., Hearnshaw, J., et al. 2014b, IJA, doi:http://dx.doi.org/10.1017/S1473550414000081

Evans, N. W. 2014, ArXiv e-prints, arXiv:1409.1788

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, Publications of the Astronomical Society of the Pacific, 125, pp. 306

Fraser, W. C., Trujillo, C., Stephens, A. W., et al. 2013, ApJL, 774, L18

Gaikwad, A., & Toke, I. M. 2009, in Proceedings of the 2Nd Workshop on High Performance Computational Finance, WHPCF '09 (New York, NY, USA: ACM), 6:1–6:9

Gaudi, S. 2011, in 2011 Sagan Exoplanet Summer Workshop

Gelman, A., & Rubin, D. 1992, Statistical science, 7, 457

Gibbs, J. W. 1902, Elementary Principles In Statistical Mechanics (Pale Bisentennial Publications)

Glicenstein, J.-F. 2001, in Astronomical Society of the Pacific Conference Series, Vol. 239, Microlensing 2000: A New Era of Microlensing Astrophysics, ed. J. W. Menzies & P. D. Sackett, 28

Goodman, J., & Weare, J. 2010, Communications in Applied Mathematics and Computational Science, 5, 65

Gould, A. 2004, ApJ, 606, 319

—. 2008, ApJ, 681, 1593

Gould, A., & Gaucherel, C. 1997, ApJ, 477, 580

Gould, A., Udalski, A., An, D., et al. 2006, ApJ, 644, L37

Gould, A., Dong, S., Gaudi, B. S., et al. 2010, ApJ, 720, 1073

Gould, A., Udalski, A., Shin, I.-G., et al. 2014, Science, 345, 46

Griest, K., & Hu, W. 1992, ApJ, 397, 362

Hamolli, L., Hafizi, M., & Nucita, A. A. 2013, International Journal of Modern Physics D, 22, 50072

Han, C., & Gould, A. 1997, ApJ, 480, 196

Harris, M. 2007, NVIDIA Developer Technology

Harris, M., Chris, Haines, K., & Staveley-Smith, L. 2008, Experimental Astronomy, 22, 129, 10.1007/s10686-008-9114-9

Hearnshaw, J. B., Abe, F., Bond, I. A., et al. 2006, in The 9th Asian-Pacific Regional IAU Meeting, ed. W. Sutantyo, P. W. Premadi, P. Mahasena, T. Hidayat, & S. Mineshige, 272

Hecht, E. 2002, Optics, 4th edn. (Addison Wesley)

Heinrich, H., Ziegenhein, P., Kamerling, C. P., Froening, H., & Oelfke, U. 2014, Journal of Physics: Conference Series, 489, 012050

Henderson, C. B., Gaudi, B. S., Han, C., et al. 2014, ApJ, 794, 52

Heyrovský, D., & Sasselov, D. 2000, ApJ, 529, 69

Hogg, D. W., Bovy, J., & Lang, D. 2010, ArXiv e-prints, 1, 1

Ingrosso, G., Novati, S. C., de Paolis, F., et al. 2011, General Relativity and Gravitation, 43, 1047

—. 2009, MNRAS, 399, 219

Isborn, C. M., Luehr, N., Ufimtsev, I. S., & Martinez, T. J. 2011, Journal of Chemical Theory and Computation, 7, 1814

Jaroszyński, M. 2002, Acta Astronomica, 52, 39

Jaroszyński, M., Udalski, A., Kubiak, M., et al. 2004, Acta Astronomica, 54, 103

Jones, M. I., Jenkins, J. S., Rojo, P., Melo, C. H. F., & Bluhm, P. 2014, ArXiv e-prints, arXiv:1409.7429

Jung, Y. K., Udalski, A., Sumi, T., et al. 2014, ArXiv e-prints, arXiv:1407.7926

Kayser, R., Refsdal, S., & Stabell, R. 1986, A&A, 166, 36

Kipping, D. M., Torres, G., Buchhave, L. A., et al. 2014, ApJ, 795, 25

Kiraga, M., & Paczyński, B. 1994, ApJ, 430, L101

Klinglesmith, D. A., & Sobieski, S. 1970, AJ, 75, 175

Klöckner, A., Pinto, N., Lee, Y., et al. 2009, CoRR, abs/0911.3456

Konacki, M., Maciejewski, A. J., & Wolszczan, A. 1999, ApJ, 513, 471

Kubas, D., Cassan, A., Dominik, M., et al. 2008, A&A, 483, 317

Kubas, D., Beaulieu, J. P., Bennett, D. P., et al. 2012, A&A, 540, A78

Kuiper, G. P. 1951, Proceedings of the National Academy of Science, 37, 1

Liebes, S. 1964, Physical Review, 133, 835

Liebig, C., & Wambsganss, J. 2010, A&A, 520, A68

Lissauer, J. J., Dawson, R. I., & Tremaine, S. 2014, Nature, 513, 336

Lissauer, J. J., Marcy, G. W., Bryson, S. T., et al. 2014, ApJ, 784, 44

Liu, J. S. 2008, Monte Carlo Strategies in Scientific Computing (Springer)

Madhusudhan, N., Crouzet, N., McCullough, P. R., Deming, D., & Hedges, C. 2014, ApJ, 791, L9

Manduca, A., Bell, R. A., & Gustafsson, B. 1977, A&A, 61, 809

Mao, S., & Paczyński, B. 1991, ApJ, 374, L37

Mashimo, T., Fukunishi, Y., Kamiya, N., et al. 2013, Journal of Chemical Theory and Computation, 9, 5599

Matsuo, T., Shibai, H., Ootsubo, T., & Tamura, M. 2007, ApJ, 662, 1282

Mayor, M., Pepe, F., Queloz, D., et al. 2003, The Messenger, 114, 20

Metropolis, N., Rosenbluth, A., Rosenbluth, M., et al. 1953, The journal of chemical physics, 21, 1087

Michalakes, J., & Vachharajani, M. 2008, in Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium, 1 –7

Milne, E. A. 1921, MNRAS, 81, 361

Mizuno, H. 1980, Progress of Theoretical Physics, 64, 544

Mizuno, H., Nakazawa, K., & Hayashi, C. 1978, Progress of Theoretical Physics, 60, 699

Mordasini, C., Alibert, Y., Benz, W., & Naef, D. 2008, in Astronomical Society of the Pacific Conference Series, Vol. 398, Extreme Solar Systems, ed. D. Fischer, F. A. Rasio, S. E. Thorsett, & A. Wolszczan, 235

Moutou, C., Deleuil, M., Guillot, T., et al. 2013, Icarus, 226, 1625

Nagatake, T., & Kunugi, T. 2010, IOP Conference Series: Materials Science and Engineering, 10, 012024

Nelder, J. A., & Mead, R. 1965, The Computer Journal, 7, 308

Newton, I. 1704, Opticks: or, A treatise of the reflections, refractions, inflections and colours of light, 1st edn. (Printers to the Royal Society, at the Prince's Arms in St. Paul's Church-yard)

Niedzielski, A., Villaver, E., Wolszczan, A., et al. 2014, ArXiv e-prints, arXiv:1410.5971

NVIDIA. 2012, NVIDIA CUDA C Programming Guide, 4th edn.

Paczyński, B. 1986a, ApJ, 301, 503

—. 1986b, ApJ, 304, 1

—. 1991, ApJL, 371, L63

—. 1996, An. Ref. A&A, 34, 419

Patel, S. G., Vogt, S. S., Marcy, G. W., et al. 2007, ApJ, 665, 744

Pejcha, O., & Heyrovský, D. 2009, ApJ, 690, 1772

Perryman, M. 2011, The Exoplanet Handbook (Cambridge University Press), cambridge Books Online

Poleski, R., Udalski, A., Dong, S., et al. 2014, ApJ, 782, 47

Poleski, R., Skowron, J., Udalski, A., et al. 2014, ApJ, 795, 42

Pollack, J. B., Hubickyj, O., Bodenheimer, P., et al. 1996, Icarus, 124, 62

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992a, Numerical Recipes Example Book [C], 2nd edn. (Cambridge University Press)

—. 1992b, Numerical Recipes in C., 2nd edn. (Cambridge University Press)

Rahvar, S., & Dominik, M. 2008, in Proceedings of the Manchester Microlensing Conference: The 12th International Conference and ANGLES Microlensing Workshop, ed. N. R. E. Kerins, S. Mao & L. Wyrzykowski

Rahvar, S., & Dominik, M. 2009, MNRAS, 392, 1193

Rattenbury, N. J., Bond, I. A., Skuljan, J., & Yock, P. C. M. 2002, MNRAS, 335, 159

Refsdal, S. 1964, MNRAS, 128, 295

Rowe, J. F., Bryson, S. T., Marcy, G. W., et al. 2014, ApJ, 784, 45

Rybicki, K., & Wyrzykowski, Ł. 2014, Acta Astron., 64, 65

Ryu, Y.-H., Han, C., Hwang, K.-H., et al. 2010, ApJ, 723, 81

Sainio, J. 2010, Computer Physics Communications, 181, 906

Sako, T., Sekiguchi, T., Sasaki, M., et al. 2008, Experimental Astronomy, 22, 51

Schneider, P., Ehlers, J., & Falco, E. E. 1992, Gravitational Lenses (Springer)

Schneider, P., & Weiss, A. 1987, A&A, 171, 49

Schramm, T., & Kayser, R. 1987, A&A, 174, 361

Seager, S., & Mallén-Ornelas, G. 2003, ApJ, 585, 1038

Shin, I.-G., Udalski, A., Han, C., et al. 2011, ApJ, 735, 85

Simpson, T. 1740, Essays on Several Curious and Useful Subjects, In Speculative and Mix'd Mathematicks (printed by H.Woodfall), 81–86

Sivia, D. 2006, Data Analysis A Bayesian Tutorial, 2nd edn. (Oxford Science Publications)

Skowron, J., & Gould, A. 2012, ArXiv e-prints, arXiv:1203.1034

Skowron, J., Udalski, A., Gould, A., et al. 2011, ApJ, 738, 87

Soszyński, I., Udalski, A., Poleski, R., et al. 2012, Acta Astron., 62, 219

Street, R. A., Choi, J.-Y., Tsapras, Y., et al. 2013, ApJ, 763, 67

Strohmaier, E. 2013, Top500 - Supercomputers, Tech. rep., Lawrence Berkeley National Laboratory, http://s.top500.org/static/lists/2013/11/PressRelease201311.pdf

Sumi, T., Kamiya, K., Bennett, D. P., et al. 2011, Nature, 473, 349

Szymański, M., Udalski, A., Kaluzny, J., et al. 1993, in Liege International Astrophysical Colloquia, Vol. 31, Liege International Astrophysical Colloquia, ed. J. Surdej, D. Fraipont-Caro, E. Gosset, S. Refsdal, & M. Remy, 503

Thompson, A. C., Fluke, C. J., Barnes, D. G., & Barsdell, B. R. 2010, New Astronomy, 15, 16

Tsapras, Y., Street, R., Horne, K., et al. 2009, Astronomische Nachrichten, 330, 4

Tsapras, Y., Choi, J.-Y., Street, R. A., et al. 2014, ApJ, 782, 48

Udalski, A. 2003, Acta Astron., 53, 291

Udalski, A., Szymanski, M., Kaluzny, J., Kubiak, M., & Mateo, M. 1992, Acta Astron., 42, 253

Udalski, A., Szymanski, M., Kaluzny, J., et al. 1994, Acta Astron., 44, 227

van Hamme, W. 1993, AJ, 106, 2096

Venot, O., & Agúndez, M. 2014, Experimental Astronomy, arXiv:1406.6566

von Soldner, J. G. 1804, Astronomisches Jahrbuch, 29, 161

Wade, R. A., & Rucinski, S. M. 1985, A&A Suppl. Ser., 60, 471

Wambsganss, J., Paczynski, B., & Schneider, P. 1990, ApJ, 358, L33

Wambsganss, J., Witt, H. J., & Schneider, P. 1992, A&A, 258, 591

Watkins, L. L., van de Ven, G., den Brok, M., & van den Bosch, R. C. E. 2013, MNRAS, 436, 2598

Wittenmyer, R. A., Tuomi, M., Butler, R. P., et al. 2014, ApJ, 791, 114

Wolszczan, A. 1994a, Science, 264, 538

—. 1994b, Astrophysics of Space and Science, 212, 67

Wolszczan, A. 2008, in Astronomical Society of the Pacific Conference Series, Vol. 398, Extreme Solar Systems, ed. D. Fischer, F. A. Rasio, S. E. Thorsett, & A. Wolszczan, 3

Wolszczan, A., & Frail, D. A. 1992, Nature, 355, 145

W.R. Gilks, S. Richardson, D. S. 1997, Markov Chain Monte Carlo In Practice (Chapman & Hall)

Zhou, G., & Huang, C. X. 2013, ApJL, 776, L35